

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA UNIVERSITARIA DE INGENIERÍA
TÉCNICA DE TELECOMUNICACIÓN**



**Instrumentación virtual para caracterización de la
fiabilidad de células solares de concentración**

**PROYECTO FIN DE MÁSTER EN INGENIERÍA DE SISTEMAS Y SERVICIOS
ACCESIBLES PARA LA SOCIEDAD DE LA INFORMACIÓN**

Sonia Linio Bragado, Julio 2012



Dedicado:

A Dioni

Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo.

Benjamin Franklin (1706-1790).

Agradecimientos:

A Dioni, por ser un apoyo constante, ayudarme a compatibilizar el trabajo, la familia, las clases y el laboratorio. Por no tener ocio por tener que estudiar o trabajar y por toda la paciencia que ha tenido que tener.

A mis padres y al enano, por haberme dado la oportunidad de llegar hasta aquí, y sobre todo por su paciencia.

A Eva, esa compañera de fatigas que desde las incontables horas estudiando esa física atravesada, ha sido un apoyo constante, y tras terminar la carrera aún nos quedaron fuerzas para embarcarnos en esta nueva aventura, sin ella no hubiera sido posible.

A Manuel y Neftalí por darme la oportunidad de trabajar con ellos, al IES por ofrecerme un laboratorio donde trabajar, a Jesús por su paciencia y a Carlos, Pilar Miguel y Vincenzo por su colaboración.



Índice

Índice	i
Índice de figuras	xi
Índice de tablas	xxii
Índice de ecuaciones	xxiii
Resumen	xxviii
Summary	xxx
1 Introducción	1
2 La célula solar	4
2.1. Introducción.	5
2.2. La célula solar fotovoltaica	6
2.3. Eficiencia	8
2.4. Circuito equivalente	9
2.5. Células solares Multiunión de semiconductores III-V.	11
3 Fiabilidad	19
3.1. Definición de fiabilidad	20
3.2. Fallos y sus clases	21
3.2.1. Fallos catastróficos y fallos por degradación	21
3.2.2. Modos, mecanismos y causas del fallo	22
1.1.1. Tipos de fallo	22
1.1.2. Fallos primarios y secundarios.	23

3.3. La fiabilidad como función del tiempo	24
3.3.1. Introducción	24
3.3.2. Fiabilidad e in fiabilidad	24
3.3.3. Distribución de fallos	25
3.3.4. Función densidad de probabilidad de fallos	25
3.3.5. Vida media	26
3.3.6. Tasa de fallo	26
3.3.7. La vida media en función de la fiabilidad	27
3.3.8. Representación gráfica	28
3.4. Funciones de distribución estadísticas de fallos	30
3.4.1. Introducción	30
3.4.2. La curva de bañera	30
3.4.3. Período infantil	32
3.4.4. La distribución exponencial	33
3.4.5. La distribución normal	36
3.4.6. La distribución log-normal	39
3.4.7. La distribución de Weibull	43
1.1.1. Características generales	43
1.1.2. Representación de los modos de fallo mediante la distribución de weibull	46
1.1.3. Análisis de Weibull	47
Resolución gráfica.	48

3.4.8. Distribución Gamma	52
3.4.9. Fallos por degradación de los componentes	53
3.5. Ensayos de fiabilidad	54
3.5.1. Introducción	54
3.5.2. Clases de ensayos de fiabilidad	54
1.1.1. Ensayos de medición y demostración	54
1.1.2. Ensayos de variación de parámetros	56
1.1.3. Ensayos de investigación de modos y mecanismos de fallo	56
1.1.4. Ensayos de cribado	57
3.5.3. Ensayos de vida de larga duración	58
1.1.1. Hipótesis de distribución	59
1.1.2. Esfuerzos y condiciones ambientales	59
3.5.4. Ensayos acelerados	60
3.5.5. Introducción a la ecuación de Arrhenius	61
4 Sistema de medida	63
4.1. Metodología de trabajo	64
4.2. Condiciones de funcionamiento	65
4.3. Potencia generada por la célula	67
5 Diagrama general y fases	70
5.1. Metodología de funcionamiento	71
5.2. Esquema	72
5.2.1. Estado de reposo	74

5.2.2. Estado Stress	75
5.2.3. Estado Medida	76
6 Esquema Hardware	78
6.1. Descripción del Hardware	79
6.1.1. Ordenador	79
6.1.2. GPIB	79
1.1.1. Descripción de GPIB	80
1.1.2. Estructura de un sistema GPIB	82
1.1.3. Características eléctricas del GPIB	83
1.1.4. Líneas de datos	84
1.1.5. Líneas de control de transferencia de datos (handshake)	84
1.1.6. Líneas de control general de la interfaz	84
1.1.7. Especificaciones mecánicas	85
1.1.8. Funcionamiento. Transferencia de datos y cronograma	86
1.1.9. Controlador GPIB HS USB	88
6.1.3. Relés OMROM G5LA	89
6.1.4. Tarjeta 16 relés	92
6.1.5. Tarjeta de adquisición de datos NI USB 6501	94
6.1.6. Concentrador USB	97
6.1.7. Fuente de Corriente 1A.	97
6.1.8. Fuente de corriente ELN 60 3,2A	99
6.1.9. Keithley 3706	100

6.1.10. Tarjeta multiplexora 3721 Dual 1×20	102
6.1.11. Fuente de alimentación Agilent 6631B	105
6.1.12. Cámara climática	106
6.1.13. Sonda de temperatura.	107
6.1.14. Alimentación	109
6.1.15. Relés HF102F	110
7 Lenguaje de programación G	111
7.1. LabView	112
7.2. Introducción al Lab-VIEW	114
7.3. ¿Cómo se programa en LabVIEW?	120
7.4. VIs y estructuras de LabVIEW usadas	125
7.4.1. Estructuras de control	125
1.1.1. Estructuras de control usadas	125
1.1.2. Colas	127
1.1.3. Librería NI-DAQmx	129
1.1.4. Librería VISA	129
1.1.5. Archivos: binarios, texto, Datalog, y TDMS.	130
1.1.6. VIs propios	133
1.1.7. Máquinas de estados	133
1.1.8. Control de una máquina de estados	136
1.1.9. Productor-consumidor	137
1.1.10. Combinaciones	139

7.4.2. Buenas prácticas de programación con LabVIEW	139
8 Arquitectura Software	142
8.1. Diagrama de estados.	143
8.1.1. Primera aproximación	143
8.1.2. Segunda aproximación	144
8.2. Gestores	145
8.2.1. Gestor de usuario	146
8.2.2. Gestor de Test	147
8.2.3. Gestor Mensajes	147
8.2.4. Gestor Enviar	148
8.2.5. Gestor USB	148
8.2.6. Gestor Fuente de alimentación	148
8.2.7. Gestor Switcher	149
8.3. Comunicación Gestores	149
9 Sistema de Instrumentación	153
9.1. Gestor Usuario	154
9.2. Estados Gestest	163
9.2.1. Funcionamiento del Gestor de Test	163
9.2.2. Reposo	165
9.2.3. PosStress	166
9.2.4. Stress	166
9.2.5. ApagarFuente	167

9.2.6. Medida	167
9.2.7. InicioAgile	168
9.2.8. InicioKeithley	168
9.2.9. SelPt100	169
9.2.10. Medir Pt100	169
9.2.11. SelDiodo	172
9.2.12. SelSwitcher	172
9.2.13. NivelFuente	174
9.2.14. EstablecerMedirI	175
9.2.15. MedirVKeithley	176
9.2.16. AlmacenarMedidas	176
9.2.17. Abrir todos	176
9.2.18. SelSigDiodo	177
9.2.19. CerrarKeithley	178
9.2.20. CerrarAgile	178
9.3. Gestor USB	179
9.3.1. PosicionStress	179
9.3.2. SelPosicion	179
9.3.3. Relés NC	180
9.3.4. ApagarFuenteCorriente	180
9.3.5. EncenderFuenteCorriente	181
9.3.6. Salir	182

9.4. Gestor Switcher	183
9.4.1. Inicializar	183
9.4.2. Abrir Canal	183
9.4.3. Abrir todos	184
9.4.4. Cerrar Canal	184
9.4.5. Visualizar Cerrados	184
9.4.6. Medir V	185
9.4.7. Medir R	185
9.4.8. Medir Termocouple	186
9.4.9. Escribir Leer IO	186
9.4.10. Cerrar	187
9.4.11. Salir	187
9.5. Gestor alimentación.	188
9.5.1. Inicio	188
9.5.2. Nivel Fuente	189
9.5.3. Medir	189
9.5.4. Cerrar	189
9.5.5. Salir	190
9.6. Controles definidos.	190
9.6.1. Control test2.ctf	190
9.6.2. Control switcher.ctf	192
9.6.3. DiodosMedir.ctf	193

9.7. Descripción VI desarrollados	194
9.7.1. ControlEstadosSwitcher.vi	194
9.7.2. ControlEstadosPuertos.vi	197
9.7.3. NumTarjetas.vi	198
9.7.4. Escribir_cabecera.vi	199
9.7.5. Guardar_cabecera.vi	200
9.7.6. Usb_all.vi	202
9.7.7. USB_cell.vi	203
9.7.8. Usb.vi	204
9.7.9. ApagarFuentes.vi	205
9.7.10. EncenderFuentes.vi	206
9.7.11. minMax.vi	208
9.7.12. Maxmin.vi	209
9.8. Variables globales	210
10 Ensayos acelerados. Puesta en marcha.	215
10.1. Primeras pruebas	216
10.2. Instalación definitiva	221
10.3. Esquemas de montaje	229
10.4. Resultados preliminares	238
10.4.1. Mecanismo de degradación y fallo	239
10.4.2. Análisis preliminar de fiabilidad	240
11 Conclusiones	243



12 Bibliografía



247

Índice de figuras

Figura 1: Unión p-n del semiconductor.....	7
Figura 2: Circuito equivalente de la célula solar y el mismo con R serie y paralelo	10
Figura 3: Espectro solar.....	12
Figura 4 : Aprovechamiento espectro solar de célula con triple unión	13
Figura 5: Sección de una célula multiunión GaInP/GaAs/Ge de las empleadas en el IES.....	14
Figura 6: Espectro solar que aprovecha cada uno de los materiales que componen las células GaInP/GaAs/Ge	15
Figura 7: Esquema de un prototipo de concentración fotovoltaica	16
Figura 8 Célula solar con concentrador.....	17
Figura 9: Distribución de fallos a lo largo del tiempo	23
Figura 10: Representación gráfica general de los parámetros de fiabilidad siendo $Q(t)=F(t)$	26
Figura 11: Función de densidad de fallo.....	28
Figura 12: Curva infiabilidad $F(t)$ y Fiabilidad $R(t)$	29
Figura 13: Función de tasa de fallo	29
Figura 14: Curva de bañera.....	31
Figura 15: Distribución exponencial.....	33
Figura 16: Distribución normal.....	37
Figura 17 Curvas de densidad de probabilidad, distribución de probabilidad	39
Figura 18: Comparación entre una distribución normal y una lognormal con los mismos percentiles del 5% y 50%. (Distribución normal normalizada a 1, distribución lognormal con el mismo factor)	40
Figura 19: Función densidad de probabilidad para distintos valores de s (0,4 a 1,4) .	42

Figura 20: Comparación entre la distribución lognormal y las distribuciones normal y exponencial	42
Figura 21: Variación de la densidad de probabilidad $f(t)$, tasa de fallos $I(t)$ y la función acumulativa de fallos $F(t)$ en función del tiempo para distintos valores del parámetro de forma β	46
Figura 22: Muestra del papel de Weibull	48
Figura 23: Lectura de los parámetros η y β en el papel de Weibull	49
Figura 24: Ejemplo representación Weibull	51
Figura 25: Distribución Gamma	52
Figura 26: Deriva de un parámetro.....	53
Figura 27: Curva en oscuridad y en iluminación para una célula solar.	68
Figura 28: Modelo para calcular la curva I-V en iluminación a partir de la curva en oscuridad.....	69
Figura 29 : Esquema general de conexionado	72
Figura 30 : Detalle de conexionado para 1 célula.	73
Figura 31 : Esquema de conexionado alimentación.....	73
Figura 32 : Detalle estado reposo.....	74
Figura 33 : Estado de reposo	74
Figura 34: Alimentación apagada para reposo.....	75
Figura 35 : Detalle estado stress	75
Figura 36 : Alimentación fuente de corriente	76
Figura 37 : Esquema circuito posición stress	76
Figura 38 : Ejemplo Posición Medida célula 1	77
Figura 39: Posición alimentación ejemplo medida célula 1	77
Figura 40: Ordenador.	79
Figura 41 : Evolución GPIB	81
Figura 42 : Esquema conexionado GPIB	82

Figura 43 : Cable de conexión GPIB: aspecto físico.	83
Figura 44: Cable de conexión GPIB: distribución de señales.....	83
Figura 45: Diagrama de tiempos de operación.....	87
Figura 46 : Conexión Gpib usb	88
Figura 47: Controlador GPIB USB HS	89
Figura 48: Omrom G5LA	89
Figura 49: Tarjeta relés.....	93
Figura 50: Esquema tarjeta relés.....	93
Figura 51: NI USB 6501.....	94
Figura 52: Labview NI 6501	95
Figura 53: Diagrama pines Ni USB 6501.....	95
Figura 54: Interconexión con tarjeta relés	96
Figura 55 : Concentrador USB	97
Figura 56 : Esquema Fuente Corriente 1 Amperio	97
Figura 57: Montaje fuentes de 1A.....	98
Figura 58: Primera simulación de conexionado.....	99
Figura 59: Fuente de corriente	99
Figura 60: Circuito fuente de corriente	100
Figura 61: Interior de la fuente de corriente.....	100
Figura 62: Dimensiones fuente de corriente.....	100
Figura 63: Keithley 3706.....	101
Figura 64: Keithley 3721	102
Figura 65 : Esquema pines tarjeta 3721	103
Figura 66: Capacidad de medida.....	103
Figura 67: Conexionado interno	105
Figura 68: Agilent 6631B	105
Figura 69 Cámara climática.....	106

Figura 70: Esquema de la estufa.....	107
Figura 71: Pt100	109
Figura 72: Relé HF102F	110
Figura 73: Pantalla de inicio de LabVIEW	116
Figura 74: Panel frontal LabVIEW	117
Figura 75 Diagrama de bloques	118
Figura 76: Panel frontal y diagrama de bloques	118
Figura 77: MAX (Measurement&Automation eXplorer).	120
Figura 78: Aspecto inicial de un VI.	121
Figura 79: Elaboración de la interfaz de usuario (panel frontal).	122
Figura 80: Elaboración del código fuente (diagrama de bloques).	123
Figura 81: VI de ejemplo donde se ve: el panel frontal, el diagrama de bloques, el conector, el icono y comentarios.	123
Figura 82 : Diseño del icono.	124
Figura 83 : Conexión de terminales.....	124
Figura 84 : Bucle While.....	125
Figura 85 : Bucle for.	126
Figura 86 : Case.	126
Figura 87 : Event	126
Figura 88. Estructuras FIFO y LIFO.	128
Figura 89. Funciones para operar con colas.	128
Figura 90. Funciones DAQmx.	129
Figura 91. Funciones VISA.....	130
Figura 92. Funciones para tratar archivos binarios y de texto.....	131
Figura 93. Funciones Datalog.....	132
Figura 94. Funciones TDMS.....	132
Figura 95: Estructura genérica de un archivo TDMS.....	133

Figura 96. Diagrama de bolas de una máquina de estados de ejemplo.	134
Figura 97. Patrón de una máquina de estados estándar.....	135
Figura 98. Patrón de una máquina de estados mediante colas.	136
Figura 99. Patrón de una máquina de estados conducida por eventos.	136
Figura 100. Ejemplo patrón de un productor-consumidor.	138
Figura 101: Clasificación general de los paradigmas de programación.	141
Figura 102: Diagrama de estados, primera aproximación.....	143
Figura 103 : Diagrama de estados, segunda aproximación.	144
Figura 104 : Estructura Gestores.....	146
Figura 105 : Gestor usuario	146
Figura 106 : Gestor test.	147
Figura 107 : Gestor Mensajes	147
Figura 108 : Gestor enviar	148
Figura 109 : Gestor USB	148
Figura 110 : Gestor Fuente.....	149
Figura 111 : Gestor switcher.....	149
Figura 112 : Estructura Gestor.	150
Figura 113 : Enviar.vi	150
Figura 114 : GEScolas.vi	151
Figura 115 : Estructura comando.	151
Figura 116 : Gestor Mensajes	152
Figura 117 : Control Test Gestor Usuario	155
Figura 118: Temporización	155
Figura 119: Evolución	156
Figura 120 : Selección canales gestor usuario	157
Figura 121: Modelo gestor usuario.....	157
Figura 122: Inicialización colas.....	158

Figura 123 : Inicia Gestores.....	159
Figura 124 : Control horario.....	159
Figura 125 : Gestor eventos- Inicio.....	160
Figura 126: Gestor eventos- Cargar.....	160
Figura 127: Gestor eventos: Guardar.	161
Figura 128: Evolución.	161
Figura 129: Actualización.	162
Figura 130 Error.....	162
Figura 131: Diagrama de estados GesTest.....	164
Figura 132: Estados.....	164
Figura 133: Detalle Colas.	165
Figura 134 : Reposo-> Iniciar	165
Figura 135 : PosStress.	166
Figura 136 : Stress	167
Figura 137 : Apagar fuente de corriente	167
Figura 138: Medida.....	168
Figura 139 : InicioAgile	168
Figura 140: Inicio Keithley	169
Figura 141: Seleccionar Pt100	169
Figura 142: Medir temperatura	170
Figura 143 : Máquina estados selección célula.....	171
Figura 144 : SelDiodo	172
Figura 145 : SelSwitcher.....	173
Figura 146 : Diagrama medida curva IV	174
Figura 147 : Nivel Fuente	175
Figura 148 : EsablecerMedirl.....	175
Figura 149 : MedirVKeithley	176

Figura 150: Almacenar medidas.....	176
Figura 151: Abrir todos	177
Figura 152 : SelSigDiodo.....	177
Figura 153: Sel Sig Diodo fin.....	178
Figura 154: Cerrar Keithley.....	178
Figura 155: Cerrar Agilent	178
Figura 156 : Posicion Stress.....	179
Figura 157: SelPosicion.....	180
Figura 158: Relés en posición NC.....	180
Figura 159: Apagar fuente de corriente	181
Figura 160: Esquema conexión fuente corriente.....	181
Figura 161 : Fuente corriente encendida.....	182
Figura 162: Encender Fuente Corriente	182
Figura 163: Salir.....	182
Figura 164: Inicializar.....	183
Figura 165: Abrir Canal.....	183
Figura 166: Abrir todos	184
Figura 167:Cerrar Canal	184
Figura 168: Visualizar Cerrados	185
Figura 169: Medir V	185
Figura 170: Medir R.....	186
Figura 171: Medir Termocouple.....	186
Figura 172: Escribir Leer IO.....	187
Figura 173: Cerrar	187
Figura 174: SALIR	188
Figura 175: Inicio	188
Figura 176: Nivel fuente.....	189

Figura 177: Medir.....	189
Figura 178: Cerrar	190
Figura 179: Salir	190
Figura 180: Control test2.ctl.....	191
Figura 181: Definición del control	191
Figura 182: Type Def control estados.....	191
Figura 183: Definición de estados	192
Figura 184: Control switcher.ctl	192
Figura 185: Type Def control switcher	193
Figura 186: DiodosMedir.ctl.....	193
Figura 187: Type Def DiodosMedir.....	194
Figura 188 : ControlEstadosSwitcher.vi.....	194
Figura 189 : Paneles Control Estados	195
Figura 190 : Medida a 2 hilos	195
Figura 191: Interconexión tarjeta canales 21 al 40.....	196
Figura 192 : Medida a 4 hilos	196
Figura 193 : ControlEstadosPuertos.....	197
Figura 194: Panel frontal ControlEstadosPuertos.	197
Figura 195: Estructura puerto 0	197
Figura 196: Estructura puerto 2	198
Figura 197 : NumTarjetas.vi	198
Figura 198: Conexión tarjetas.....	199
Figura 199 : escribir_cabecera.vi.....	199
Figura 200 : escribir_cabecera	200
Figura 201 : guardar cabecera	200
Figura 202 : Código guardar cabecera	201
Figura 203: Usb_all.....	202

Figura 204: Panel frontal usb_all	202
Figura 205: diagrama usb_all	203
Figura 206: Usb_cell.....	203
Figura 207: Código para abrir o cerrar un relé.	204
Figura 208: Usb.vi.....	204
Figura 209: Panel frontal usb.vi	205
Figura 210: Diagrama de bloques usb.vi	205
Figura 211: Apagar fuente	206
Figura 212 : Apagar Alimentación Fuente 4 relés con 220V	206
Figura 213: Apagar Alimentación Fuente 2Relés 5V.....	206
Figura 214: Encender Fuentes	207
Figura 215: Encender Alimentación Fuentes con 4 relés a 220V.....	207
Figura 216: Encender Fuentes con 2 relés a 5V	207
Figura 217: min Max.vi	208
Figura 218: Ecuación minMAX	208
Figura 219:MAXmin.vi	209
Figura 220: Ecuación MAXmin	209
Figura 221: Panel frontal MAXmin.....	210
Figura 222: Variables globales usuario	211
Figura 223: Variables globales medidas.....	212
Figura 224: Variables globales, tiempos.....	213
Figura 225: Variables globales Agilent	213
Figura 226: Variables globales, medida.	214
Figura 227: Variables globales, keithley	214
Figura 228: Diodos testeados.....	216
Figura 229: Interconexión tarjeta-diodos	218
Figura 230: Primer montaje tarjeta 3721	218

Figura 231: Interconexión Tarjeta 3721 con diodos	219
Figura 232 : Primer montaje con 1 tarjeta de relés.....	219
Figura 233: Pruebas	220
Figura 234: Instalación con dos tarjetas y fuentes de corriente	220
Figura 235: Detalle potenciómetros fuentes de corriente	221
Figura 236: Distribución del espacio.....	222
Figura 237: Bandeja 1	222
Figura 238: Bandeja 2	223
Figura 239: Detalle alimentación	223
Figura 240: Bandeja 1 y 2.....	224
Figura 241: Conexión bandeja relés con bornero.....	224
Figura 242: Detalle trasera bornero.....	225
Figura 243: Conexión bornero con tarjeta Keithley	226
Figura 244: Cableado bandejas.....	227
Figura 245: Detalle ventiladores	227
Figura 246: Bandeja 1 funcionando.....	228
Figura 247: Detalle tarjeta bandeja 2 funcionando	228
Figura 248: Esquema conexión tarjeta relés (2 a 5).....	232
Figura 249: Esquema conexión tarjetas 1 y 6	233
Figura 250 : Alimentación	234
Figura 251 : Curva IV caracterizada vs obtenida con labview	235
Figura 252: Curva IV minMAX vs MAXmin.....	236
Figura 253: Curva IV con tiempo de estrés	236
Figura 254: Configuración primer ensayo.....	237
Figura 255: Tiempo estrés primer ensayo	237
Figura 256: Evolución de la curva I/V de una célula solar durante el ensayo a 154°C.	240

Figura 257: Representación de Weibull de los tiempo de fallos.....	241
Figura 258: 20 células funcionando.....	245
Figura 259: Primer ensayo 20 células	245

Índice de tablas

Tabla I: Especificaciones mecánicas relé Omrom G5LA	90
Tabla II: Características relé Omrom G5LA	91
Tabla III: Equivalencia pines puerto 0	96
Tabla IV: Equivalencia pines puerto 2	97
Tabla V: Evolución de Labview	115
Tabla VI: Conexión tarjeta pruebas	217
Tabla VII: Tiempo de fallo de las diferentes células del ensayo a 154°C (las células 8 y 14 son las células sin inyección de corriente).	241

Índice de ecuaciones

$\varphi \cos 1 = AM$ (1)	5
$E = E_c - E_v$ (2)	7
$E = h\nu = \frac{h.c}{\lambda}$ (3).....	8
$I = IL - I_0 e^{EV + IR_s K T c - 1 - V + IR_s R_p}$ (4)	10
$I = I_{sc} 1 - e - e(V_{oc} - V) m K T$ (5)	11
$R(t) = \Pr(T > t)$ (6)	24
$N(0) = N_f(t) + N_s(t)$ (7).....	24
$R_t = N_s(t)N(0) = 1 - N_f(t)N(0)$ (8)	24
$F(t) = \frac{N_f(t)}{N(0)}$ (9).....	25
$F(t) = 1 - R(t)$ (10).....	25
$F(t) = P(\tau \leq t) = \int_0^t f(x)dx$ (11).....	25
$F(0) = 0, F(\infty) = 1$ (12)	25
$R(t) = 1 - F(t) = P(\tau > t) = \int_t^\infty f(x)dx$ (13).....	25
$R(0) = 1, R(\infty) = 0$ (14)	25
$f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt}$ (15)	25
$f(t) dt = R(t) \cdot \lambda(t) dt$ (16).....	26
$\theta = E(\tau) = \int_0^\infty t f(t) dt$ (17).....	26
$P(t < \tau \leq t + \Delta t \tau > t) = \frac{P(t < \tau \leq t + \Delta t)}{P(\tau > t)} = \frac{R(t) - R(t + \Delta t)}{R(t)}$ (18).....	27

$$z(t, \Delta t) = \frac{R(t) - R(t + \Delta t)}{\Delta t R(t)} \quad (19) \dots\dots\dots 27$$

$$z(t) = -\frac{1}{R(t)} \frac{dR(t)}{dt} \quad (20) \dots\dots\dots 27$$

$$\theta = \int_0^\infty t f(t) dt = -\int_0^\infty t dR(t) = \left| tR(t) \right|_0^\infty + \int_0^\infty R(t) dt \quad (21) \dots\dots\dots 27$$

$$-\lim_{t \rightarrow \infty} [tR(t)] \quad (22) \dots\dots\dots 27$$

$$\lim_{t \rightarrow \infty} [tR(t)] = \lim_{t \rightarrow \infty} \frac{t}{R^{-1}(t)} = \lim_{t \rightarrow \infty} \frac{1}{R^{-2}(t) f(t)} = \lim_{t \rightarrow \infty} \frac{R(t)}{z(t)} \quad (23) \dots\dots\dots 27$$

$$\theta = \int_0^\infty R(t) dt \quad (24) \dots\dots\dots 28$$

$$z(50) = \frac{f(50)}{R(50)} = \frac{0,0072}{0,88} = 0,0082 \quad (25) \dots\dots\dots 29$$

$$f(t) = af_1(t) + bf_2(t) + cf_3(t) \quad (26) \dots\dots\dots 31$$

$$\int_0^\infty f(t) dt = 1 \quad (27) \dots\dots\dots 31$$

$$f_1(t) = \frac{1}{100} \left(\frac{1}{50} \right)^{-0,5} e^{-\left(\frac{1}{50} \right)^{0,5}} \quad (28) \dots\dots\dots 31$$

$$f_2(t) = \frac{1}{100000} e^{-t/100000} \quad (29) \dots\dots\dots 32$$

$$f_3 = \frac{1}{1000\sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{t-10000}{1000} \right)^2} \quad (30) \dots\dots\dots 32$$

$$f(x) = \frac{1}{\beta} x^{\frac{-x}{\beta}}, \quad x > 0 \quad (31) \dots\dots\dots 33$$

$$p(0, \lambda t) = \frac{E^{-\lambda t} (\lambda t)^0}{0!}; \quad E=2.718 \quad (32) \dots\dots\dots 34$$

$$P(X \geq x) = E^{-\lambda t} \quad (33) \dots\dots\dots 34$$

$F(t) = 1 - e^{-\lambda t}$ (34).....	34
$f(t) = \lambda e^{-\lambda t}$ (35).....	34
$R(t) = e^{-\lambda t}$ (36).....	34
$z(t) = \lambda = \frac{1}{\beta}$ (37).....	35
$\theta = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t}dt = \frac{1}{\lambda}$ (38).....	35
$E^{-\lambda t} F(x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} E^{\frac{-(x-\mu)^2}{2\sigma^2}}$ (39).....	36
$p(a \geq x < b) = \int_a^b f(x, \mu, \sigma^2)dx$ (40).....	36
$z = \frac{x - \mu}{\sigma} = \text{valor}$ (41).....	37
$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\theta}{\sigma}\right)^2}$ (42).....	38
$F(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{1}{2}\left(\frac{x-\theta}{\sigma}\right)^2} dx$ (43).....	38
$R(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_t^\infty e^{-\frac{1}{2}\left(\frac{x-\theta}{\sigma}\right)^2} dx$ (44).....	38
$E(\tau) = \int_{-\infty}^\infty tf(t)dt = \theta$ (45).....	38
$V(\tau) = \int_{-\infty}^\infty (t - \theta)^2 f(t)dt = \sigma^2$ (46).....	38
$z(t) = \frac{e^{-\frac{1}{2}\left(\frac{t-\theta}{\sigma}\right)^2}}{\int_t^\infty e^{-\frac{1}{2}\left(\frac{x-\theta}{\sigma}\right)^2} dx}$ (47).....	38

$$z = \frac{t - \theta}{\sigma} \quad (48) \dots\dots\dots 39$$

$$f(t) = \frac{1}{t\sigma'\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln t - \theta'}{\sigma'}\right)^2} \quad (49) \dots\dots\dots 41$$

$$\theta = E(\tau) = e^{\theta' + \frac{\sigma'^2}{2}}$$

$$\sigma^2 = V(\tau) = e^{2\theta' + \sigma'^2} (e^{\sigma'^2} - 1) \quad (50) \dots\dots\dots 41$$

$$\lambda t = -dRtdtRt \quad (51) \dots\dots\dots 43$$

$$\lambda t dt = t - t_0 \eta \beta \quad (52) \dots\dots\dots 44$$

$$R(t) = e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (53) \dots\dots\dots 44$$

$$F(t) = 1 - e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (54) \dots\dots\dots 44$$

$$f(t) = \frac{\beta}{\eta^\beta} (t - \gamma)^{\beta-1} e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (55) \dots\dots\dots 45$$

$$z(t) = \frac{\beta}{\eta^\beta} (t - \gamma)^{\beta-1} \quad (56) \dots\dots\dots 45$$

$$R(t) = \frac{1}{1 - F(t)} = e^{\left(\frac{1}{\eta}\right)^\beta} \quad (57) \dots\dots\dots 49$$

$$\ln \ln \frac{1}{1 - F(t)} = \beta \ln t - \beta \ln \eta \quad (58) \dots\dots\dots 49$$

$$f(t) = \frac{\lambda}{\Gamma(\tau)} (\lambda t)^{\tau-1} \quad (59) \dots\dots\dots 52$$

$$F(t) = \frac{\lambda}{\Gamma(\tau)} \int_0^t (\lambda x)^{\tau-1} e^{-\lambda x} dx \quad (60) \dots\dots\dots 52$$

$$R(t) = \frac{\lambda}{\Gamma(\tau)} \int_t^\infty (\lambda x)^{\tau-1} e^{-\lambda x} dx \quad (61) \dots\dots\dots 52$$

$\theta = E(\Gamma) = \frac{\tau}{\lambda}$	(62).....	52
$z(t) = \frac{t^{\tau-1} e^{-\lambda x}}{\int_t^\infty x^{\tau-1} e^{-\lambda x} dx}$	(63).....	52
$R_D(t) = \int_{z_l}^{z_s} f(z, t) dz$	(64)	53
$Máx(R) = Máx\left(e^{-\int_{t_0}^{t_0+h} z(t) dt}\right)$	(65).....	57
$Mín\left(\int_{t_0}^{t_0+h} z(t) dt\right)$	(66).....	57
$t(T) = Ae^{\frac{E_A}{KT}};$	(67).....	61
$\ln t(T) = \ln A + \frac{E_A}{KT}$	(68)	61
$Vilum = Vdark - ILRs$	(69).....	69
$Ilum = IL - Idark$	(70)	69
$Iout = VrefR1$	(71).....	98
$R = R0.1 + \alpha \cdot \Delta T$	(72).....	108

Resumen

Este Trabajo Fin de Máster surge de la necesidad de evaluar la fiabilidad de los sistemas fotovoltaicos de concentración, los cuales han sufrido una evolución importante, pasando de usarse células solares de silicio con un 26% de rendimiento, a células multiunión III-V superando el 43% de rendimiento. Las células solares multinunión, mucho más caras y complejas que las células de silicio, no podrán ser comercializadas hasta que no se demuestre que tienen una fiabilidad comparable a las células de silicio.

Con el objetivo de disponer de resultados de fiabilidad en un periodo de tiempo adecuado, se utilizan ensayos acelerados. Los ensayos acelerados en células solares presentan una dificultad añadida, debido a la necesidad de que la célula solar esté funcionando y además sea caracterizada dentro de una cámara climática. Mientras que para realizar ensayos acelerados en otros dispositivos es muy sencillo hacerlos funcionar dentro de la cámara climática, en el caso de las células solares tanto el funcionamiento como la caracterización requieren de iluminación dentro de la cámara climática. Conseguir dicha iluminación es complejo como se comentará en el desarrollo de esta memoria de Trabajo Fin de Máster, así como la solución encontrada.

A lo largo de esta memoria se desarrollará una primera parte teórica, comenzando con una breve descripción teórica sobre células solares, aunque el estudio de las mismas no es el objetivo de este proyecto, por lo que se continuará con teoría de fiabilidad. El objetivo de este Trabajo Fin de Máster es desarrollar un software capaz de realizar ensayos acelerados sobre células solares. El diseño del software desarrollado podría usarse para cualquier tipo de célula solar, u otro dispositivo similar como un LED o un diodo láser.

El último capítulo teórico desarrollado en este proyecto es una introducción al lenguaje de programación gráfico, denominado lenguaje G, implementado con Labview, software elegido para el desarrollo del programa. Dado que estará destinado a su uso en otros proyectos, el desarrollo del sistema estará totalmente descrito y el código comentado, para que en un futuro se pueda modificar de forma sencilla.

El núcleo de la memoria es el desarrollo del software aunque también se mostrará el desarrollo hardware, que ha sido desarrollado en paralelo en otro Trabajo

Fin de Máster y la instalación necesaria para poder llevar a cabo los ensayos. Para finalizar la memoria, se documenta la instalación realizada, mostrando las pruebas realizadas al software y al hardware y la puesta en funcionamiento de los ensayos con sus primeros resultados.

Summary

This Master's Thesis arises from the need to assess the reliability of concentration photovoltaic systems, which have undergone a significant evolution, from used silicon solar cells with a 26% efficiency, a III-V multi-junction cells exceeding the a 43% efficiency. Multi-junction solar cells, much more expensive and complex than silicon cells, cannot be marketed until they demonstrate a reliability comparable to silicon cells.

With the aim of providing reliable results in an adequate period of time, there are used accelerated tests. The accelerated testing in solar cells adds a difficulty because of the need for the cell to work and be characterized moreover in a climatic chamber. While for other devices an accelerated testing is very simple to make them operate within the climate chamber, in the case of the solar cells, the evaluation of the reliability of this type of devices by means of accelerated tests is especially problematic when they work at medium or high concentration, because it is practically impossible to emulate real working conditions of the cell inside climatic chambers, the characterization require illumination within the climatic chamber. Getting such lighting is complex as discussed in the development of this Master's Thesis and the solution found.

Throughout this memory will develop a theoretical part, beginning with a short theoretical description of solar cells, although the study of this is not the objective of this project, which will continue with the theory of reliability. The aim of this Master's Thesis is to develop software capable of accelerated tests on solar cells. The design of the software developed could be used for any type of solar cell, or other similar device such as an LED or a laser diode.

The last theoretical chapter explained in this project is an introduction to graphical programming language called G language, implemented in Labview, software chosen for the development of the program. As will be for use in other projects, the development of the system is fully described and the code commented, so that in future will be easy to modify if necessary.

Main memory is focused on the software development but also shows the hardware development, which has been developed in parallel in another Master's Thesis and installation necessary to carry out the tests. To end the report, it's



documented the installation made, showing the tests made to the software and hardware, and the testing commissioning and its first results.

1 Introducción

La tecnología de las células solares está en auge, y en la actualidad existen diferentes tipos de células solares tanto por el número de uniones de que disponen como por los materiales que utilizan, debido a ello, se hace necesario evaluar la fiabilidad de las mismas. Dicha fiabilidad es imposible evaluarla en condiciones de trabajo normales, dado que se necesitarían muchos años para empezar a ver su degeneración, por lo que es necesario implementar una nueva metodología e instrumentación para realizar ensayos acelerados en células, el cual es el objetivo de este Trabajo Fin de Máster.

Los ensayos sobre células solares cuentan con una dificultad añadida, debido a la necesidad de que la célula solar esté funcionando y además sea caracterizada dentro de una cámara climática. Mientras que para realizar ensayos acelerados en otros dispositivos es muy sencillo hacerlos funcionar dentro de la cámara, en el caso de las células solares tanto el funcionamiento como la caracterización requieren de iluminación dentro de la cámara climática, conseguir dicha iluminación es complejo como se comentará en el desarrollo de este proyecto, así como la solución encontrada.

A lo largo de este Trabajo Fin de Máster se explicará con todo detalle la metodología implementada, para poder realizar los ensayos acelerados sobre células solares, simulando que éstas están en condiciones reales aceleradas de trabajo.

Las células solares para las que está destinado este proyecto son células solares de triple unión con semiconductores III-V que funcionan a elevadas concentraciones luminosas, hasta 1000 soles, pero dado su diseño, puede ser adaptado a diferentes células, puesto que las opciones de medida son configurables por el usuario en función de las necesidades del momento.

La memoria ha sido dividida en los siguientes capítulos:

La célula solar.

Fiabilidad.

Sistema de medida.

Diagrama general y fases.

Esquema hardware.

Lenguaje de programación G.

Arquitectura software.

Sistema de instrumentación.

Ensayos acelerados, puesta en marcha.

Conclusiones.

2 La célula solar

2.1. Introducción.

Una célula solar es un dispositivo semiconductor cuya función es convertir los fotones procedentes de la luz solar en electricidad. El efecto fotovoltaico convierte la energía de los fotones en energía eléctrica en continua. La naturaleza espectral de la luz solar es importante para el diseño de dispositivos fotovoltaicos o células solares, condicionando las características de los materiales aptos para estas aplicaciones. Para analizar el espectro de la radiación solar es una buena aproximación considerar que el Sol actúa como un emisor perfecto de radiación (cuerpo negro) a una temperatura cercana a los 5800 K. La constante solar es la cantidad de energía recibida del Sol por unidad de tiempo y unidad de superficie, medida fuera de la atmósfera terrestre en un plano perpendicular a los rayos a una distancia de 1 UA del Sol (1UA=distancia Sol-Tierra). Como la distancia Tierra–Sol va variando a lo largo del año, se utiliza en consecuencia como constante solar su valor medio que es de 1366,1 W/m².

Se define la masa de aire AM0 (Air Mass 0) como la ausencia de atenuación atmosférica de la irradiación solar a 1 UA del Sol. La distribución espectral AM0 difiere del caso ideal para un cuerpo negro, lo cual se debe a efectos como la variación de la transmitancia de la atmósfera solar para diferentes longitudes de onda.

Cuando la radiación pasa a través de la atmósfera de la Tierra, es atenuada en su intensidad por diversos procesos que se producen a lo largo de su recorrido a través de la misma. Estos procesos son; absorción selectiva por los gases y por el vapor de agua, dispersión molecular (o de Rayleigh) por los mismos componentes y dispersión por aerosoles (o de Mie) y absorción por los mismos. El resultado de todos estos procesos es la descomposición de la radiación solar incidente en componentes diferenciadas. En condiciones atmosféricas claras el parámetro más importante para la determinación de la potencia total es la longitud del camino que recorre la luz a través de la atmósfera. Cuando el Sol se encuentra en dirección vertical, el camino es mínimo y se puede decir que la radiación que llega a la superficie de la Tierra es AM1 (Air Mass 1). Si el ángulo respecto de la vertical es φ resulta:

$$\varphi \cos 1 = AM \quad (1)$$

La radiación solar que llega a la superficie terrestre después de atravesar la atmósfera se conoce como radiación solar directa. La radiación solar difusa es la dispersada por el aire y las nubes. Y el albedo es radiación reflejada por el suelo.

La radiación total sobre una superficie es la suma de las tres radiaciones (directa + difusa + albedo) y se conoce como radiación global o total.

2.2. La célula solar fotovoltaica

Para la fabricación de las células solares se emplean materiales semiconductores. El empleo de estos materiales se debe, a que tienen la propiedad de que la energía necesaria para separar cierto número de electrones de su núcleo, es similar a la energía de los fotones que forman la luz solar. De esta forma modificando el material semiconductor en ciertas condiciones es posible mediante el efecto fotovoltaico transformar parcialmente a energía luminosa en energía eléctrica.

El material semiconductor más utilizado en la actualidad para la fabricación de células solares es el silicio ya que es el material semiconductor más abundante. A las células solares de Silicio se les denomina de primera generación y se clasifican según la ordenación de los átomos como silicio amorfo, policristalino o monocristalino.

Una célula solar es como un diodo de gran tamaño. Para la fabricación de la célula solar es necesario construir un material semiconductor con una unión p-n. Para ello partimos de un cristal semiconductor de silicio puro se le denomina semiconductor intrínseco, al semiconductor intrínseco se le añaden de forma voluntaria impurezas, para construir semiconductores extrínsecos.

La razón por la que se añaden impurezas a un semiconductor intrínseco es para que tenga un mayor o menor número de electrones de valencia que el material sin impurezas, modificando así su conductividad. Dependiendo de cuál sea esta impureza logramos semiconductores de tipo n ó p. El silicio tiene cuatro electrones de valencia por lo que para doparlo, impurificarlo, se emplean elementos que tengan tres o cinco electrones de valencia.

En el caso del Si, si las impurezas provienen de elementos la columna V de la tabla periódica (P, As, Sb, Bi), las mismas aportan cuatro electrones para satisfacer los enlaces covalentes, dejando un quinto libre. Estos dopantes se denominan donores o dadores. El electrón libre tiene una energía de ionización muy baja (~ 0.05 eV) y, a bajas temperaturas ($\sim 4^\circ\text{K}$), permanece en órbita alrededor de su átomo. A temperaturas más altas (en particular a temperatura ambiente) se ioniza y contribuye a la conductividad eléctrica, dejando atrás el ion positivo del átomo donador. A

semiconductores dopados con elementos donores se los denomina semiconductores tipo n.

Si el dopante introducido pertenece al grupo III (B, Al, Ga, In), como estos elementos tienen tres electrones en su último nivel, no llegan a completar los cuatro enlaces con los átomos vecinos de Si, requiriéndose el atrapamiento de un electrón, que deja entonces un hueco positivo libre en la banda de valencia. Este semiconductor es de tipo p y a los átomos causantes de la conducción por huecos se los llama aceptores.

Si en un mismo material semiconductor tenemos dos zonas adyacentes, una de ellas con dopado n y la otra con dopado p tendremos una unión p-n. Esta unión p-n permite que los pares electrón-hueco fotogenerados cuando se ilumina el material puedan ser utilizados para generar una corriente eléctrica. En la siguiente figura podemos ver la unión p-n:

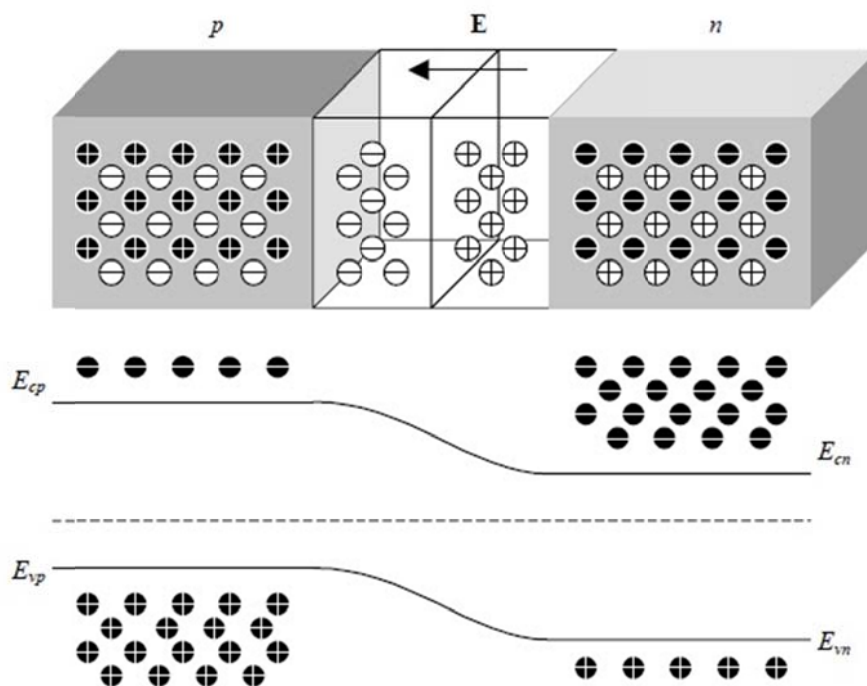


Figura 1: Unión p-n del semiconductor

Para que se fotogenera un par electrón-hueco es necesario que la energía del fotón sea mayor que la diferencia del gap:

$$E = E_c - E_v \quad (2)$$

Donde:

E_c , Energía de un electrón en la banda de conducción.

E_v , Energía de un electrón en la banda de valencia.

Un fotón tiene una energía:

$$E = h \nu = \frac{h \cdot c}{\lambda} \quad (3)$$

h : constante de Plank

ν : Frecuencia del fotón

λ : longitud de onda

c : velocidad de la luz en el medio

Solo los fotones cuya energía es mayor que la energía del gap contribuirán a la generación de corriente continua. Las células solares tienen contactos eléctricos en ambas caras para recoger la corriente eléctrica. La cara que no recibe luz se cubre totalmente para protegerla. La cara que está expuesta al sol se cubre mediante una rejilla metálica que permite recibir la mayor cantidad posible de radiación.

2.3. Eficiencia

La eficiencia de la célula se mide como el cociente entre la potencia eléctrica que suministra la célula y la potencia de la radiación que incide sobre la célula en ese instante. La eficiencia máxima alcanzada para una célula convencional ha sido de un 28%. En laboratorio se suele conseguir una eficiencia de un 25%, pero en la práctica una célula comercial de silicio alcanza una efectividad en torno a 20%. Con células de triple unión de Solar Junction se ha llegado a alcanzar el 43,5% de eficiencia, este valor es determinante por lo que se trabaja para mejorar la eficiencia.

El diseño de la célula influye en la eficiencia de la misma por lo que se intenta maximizarla en lo posible siguiendo las siguientes pautas:

- Aumento de la generación de pares e-/h+ para ello se debe aumentar la absorción de luz, se emplean materiales texturizados y capas anti reflectantes.

- Reducción de las pérdidas por recombinación, se emplean materiales de alta calidad, en fabricación se emplean técnicas para eliminar impurezas contaminantes y técnicas que reducen la actividad recombinante de defectos. También se emplean dieléctricos como el óxido de silicio o el nitruro de silicio.
- Se intentan minimizar los efectos resistivos eligiendo metales apropiados con buen contacto y poca resistencia, con silicio muy dopado la resistencia de contacto disminuye. También ha de tenerse en cuenta en el diseño de la maya de metalización el motivo y las dimensiones.
- Para aumentar la absorción de luz, se emplean materiales texturizados y capas anti reflectantes.

La célula de forma individual proporciona muy poca energía. Por ello se agrupan varias células en serie o paralelo haciéndolas funcionar como si fueran sólo una. Es lo que llama panel o módulo fotovoltaico.

Las células que forman un panel, proporcionan más o menos electricidad en función de la energía solar que incida sobre la superficie de las células. Pero además la respuesta de las células frente a la radiación solar estará condicionada a los materiales empleados en su fabricación, así como al diseño empleado.

2.4. Circuito equivalente

Para poder medir las prestaciones de una célula solar de concentración es necesario definir una serie de características. Una de las más importantes es la que se conoce como curva I-V, curva de intensidad-voltaje. Esta curva nos da información sobre los distintos valores de tensión e intensidad que puede proporcionar la célula. En capítulos posteriores veremos en más detalle la importancia de esta curva en el estudio de células solares.

El funcionamiento de la célula se puede simular por circuito equivalente constituido por un diodo de unión p-n y por un generador de corriente, de valor I_f , a los que habría que añadir una resistencia serie R_s y una resistencia paralelo R_p . La primera proviene principalmente de la resistividad del emisor n y de la base p, de las resistencias de contacto entre el semiconductor y los contactos metálicos, y de la resistencia de los dedos metálicos que constituyen la malla de la metalización frontal.

La segunda es esencialmente debida a corrientes de fugas en los bordes y a la presencia de defectos cristalinos y de impurezas en la zona de la unión.

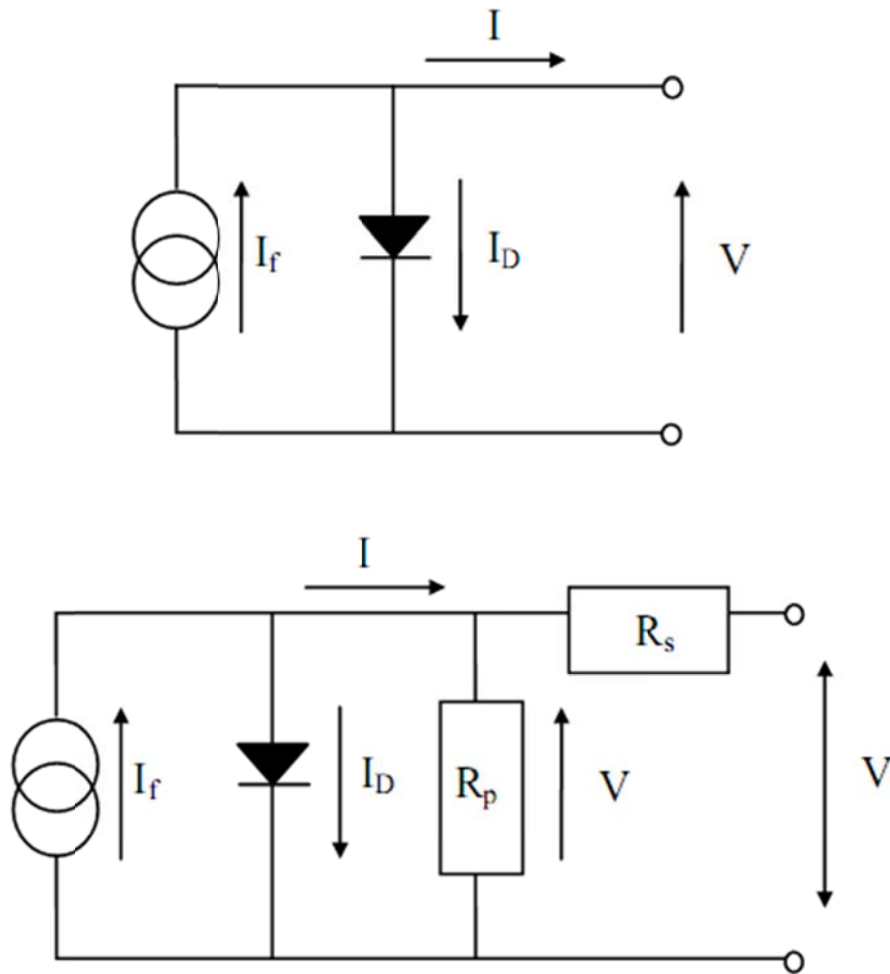


Figura 2: Circuito equivalente de la célula solar y el mismo con R serie y paralelo

Por lo tanto, la curva I/V de la célula solar se puede ajustar a la siguiente ecuación.

$$I = I_L - I_0 \left[e^{\frac{E(V+IR_s)}{KT_c}} - 1 \right] - \frac{V+IR_s}{R_p} \quad (4)$$

Siendo:

e , es la carga del electrón e igual a 1.6021×10^{-19} C.

m , es un parámetro constructivo de la célula, normalmente igual a 1.

K , es la constante de Boltzman.

T, Temperatura en °Kelvin.

IL, Densidad de corriente fotogenerada, Densidad de corriente en oscuridad.

R_s , resistencia serie global, que se compone de la resistencia ohmica a la circulación de corriente en el emisor, al paso de la corriente del emisor al contacto frontal, la circulación de corriente por la malla frontal, la circulación de la corriente en la base, capa pasivante y sustrato y al paso de corriente del sustrato al contacto metálico posterior.

R_p , resistencia paralelo, que representa las fugas de corriente debidas a los cortocircuitos locales a través de la unión p-n.

V, tensión dada para la que circulará por la célula la densidad de corriente I.

La curva I-V de la célula solar también puede representarse en función de la tensión de circuito abierto y de la corriente de cortocircuito mediante la siguiente expresión:

$$I = I_{sc} \left(1 - e^{\frac{-e(V_{oc}-V)}{mKT}} \right) \quad (5)$$

Siendo I_{sc} la corriente en cortocircuito y V_{oc} la tensión en circuito abierto, cuando no está conectada a ninguna carga.

Como ya hemos indicado anteriormente hasta ahora las células solares más usadas eran de Silicio, esto está cambiando ya que se están empezando a comercializar células solares multiunión que son mucho más eficientes. Este tipo de células son las que se han empleado en este Trabajo fin de Máster, siendo de los primeros ensayos de fiabilidad que se realizan sobre este tipo de células comerciales.

2.5. Células solares Multiunión de semiconductores III-V.

Las células solares de silicio tienen rendimientos bajos debido a su mal aprovechamiento de la radiación solar por lo que se han propuesto células solares multiunión que aprovechan la radiación solar de forma mucho más eficiente.

Uno de los objetivos a conseguir para obtener alta eficiencia es que el espectro de radiación sea lo más cercano posible al gap del semiconductor. Por este motivo se

están diseñando células con varios tipos de semiconductor que permite aprovechar casi todo el espectro solar.

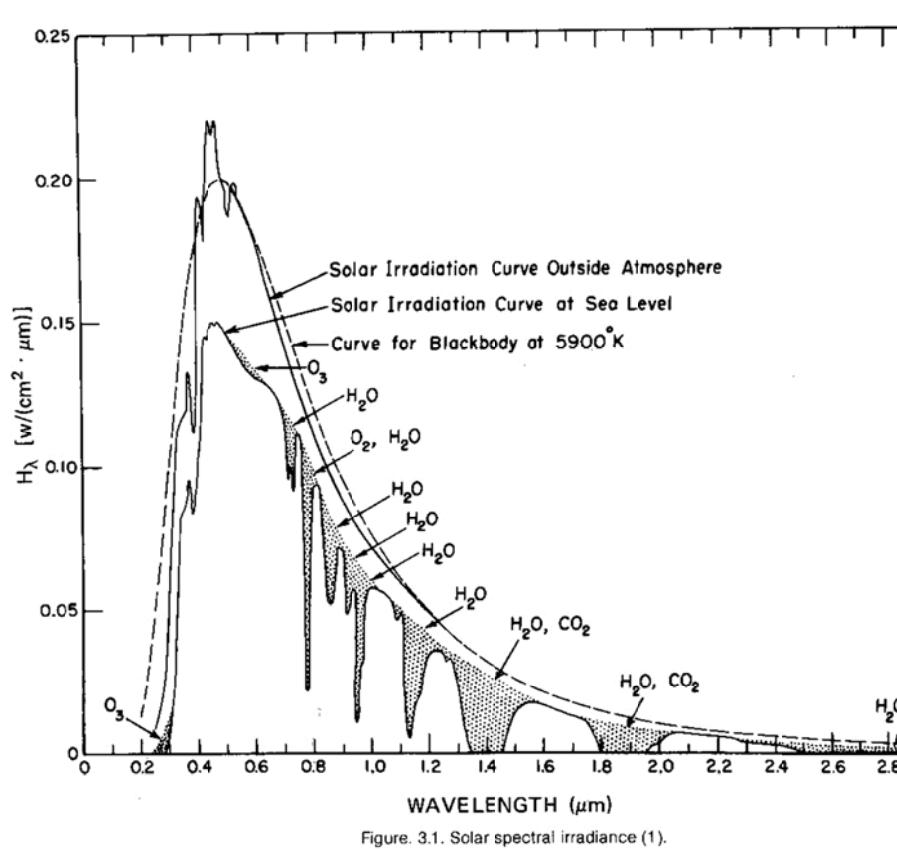


Figura 3: Espectro solar

Estas células incorporan dos o más capas de semiconductores para la absorción de diferentes longitudes de onda de la luz solar, en lugar de la única capa que se empleaba en las células convencionales de silicio. Esto permite incrementar la eficiencia. Normalmente se emplean semiconductores III-V, como el arseniuro de galio (GaAs), estas células están formados por un elemento de la columna III de la tabla periódica y un elemento de la columna V, con esto pueden obtenerse compuestos binarios como el GaAs o el InP, ternarios como el AlGaAs o el GaInP o cuaternarios como el GaInNAs o el AlGaInP por lo que se puede seleccionar el gap del material semiconductor mediante la selección de la composición del material. Las propiedades de estos semiconductores los hacen idóneos para la conversión fotovoltaica, en la siguiente figura, se muestra la comparación del aprovechamiento del espectro solar recibido por un material con una única unión y un material con tres uniones, viendo como el material con tres uniones tiene un aprovechamiento mayor del espectro solar.

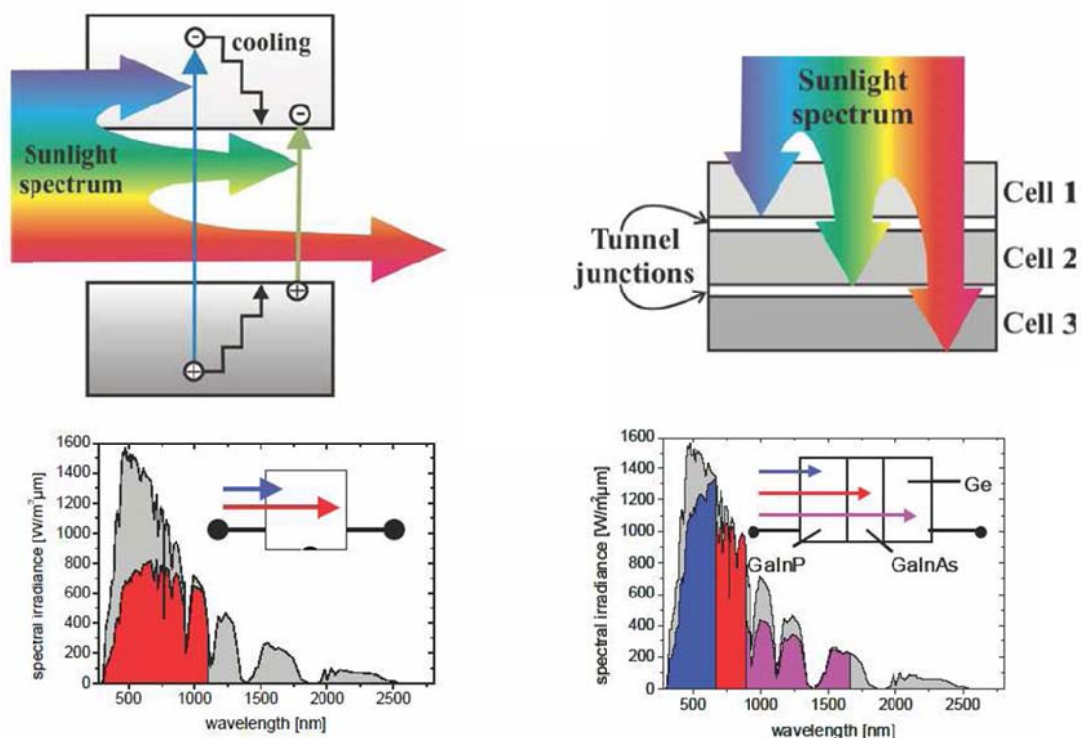


Figura 4 : Aprovechamiento espectral solar de célula con triple unión

Como podemos ver a continuación en la tabla, el silicio produce una Voc de tan solo el 58% de lo que podría producir en función de su E_g , mientras que las de GaAs llegan al 69,9%.

T = 300 K	E_g (eV)	Voc (V)	Voc/ E_g (V/eV)	V_m (V)	$qV - E_g$ (eV)	dV/dT (mV/°C)
Silicio (@ Voc)	1,12	0,580	0,580	---	0,47	- 2,09
Silicio (@ V_m)	1,12	---	---	0,55	0,57	- 2,42
GaAs (@ Voc)	1,43	1,00	0,699	---	0,43	- 1,76

Tabla. I. Parámetros para las células solares de silicio y GaAs.

Siendo:

E_g , la energía del ancho de banda prohibida.

Voc, tensión de circuito abierto

V_m , tensión en el punto de máxima potencia.

V , tensión aplicada.

T , temperatura.

Actualmente las células solares III-V se suelen diseñar en configuración multiunión, cada uno de los semiconductores que componen a este tipo de células se caracterizan por una banda de energía por lo que los hacen más sensibles a absorber una determinada región del espectro solar. Seleccionando los materiales se consigue abarcar la mayoría del espectro solar.

Los semiconductores III-V son transparentes a los fotones cuya energía sea inferior a su E_g , por lo que podremos montarlos uno sobre otro. Cada capa de semiconductor debe ser muy delgada para dejar pasar la luz, por lo que se montarán todas sobre un sustrato monocristalino único, colocando cada capa de semiconductor de mayor a menor E_g . De esta forma podremos construir células solares con multitud de materiales diferentes para aprovechar casi íntegramente el espectro solar. Teóricamente si colocásemos infinitas uniones p-n de semiconductores, podríamos alcanzar el 86,5% de eficiencia. En la realidad no es necesario colocar muchos materiales para conseguir aumentar la eficiencia respecto a las monounión.

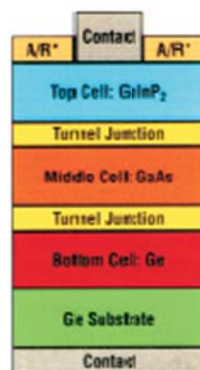


Figura 5: Sección de una célula multiunión GaInP/GaAs/Ge de las empleadas en el IES

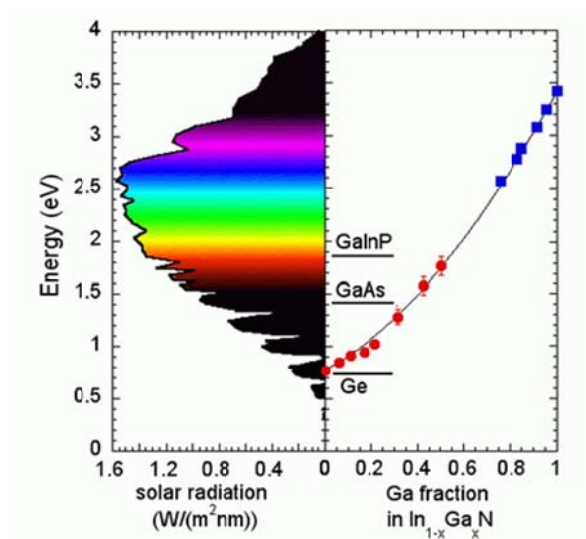


Figura 6: Espectro solar que aprovecha cada uno de los materiales que componen las células GaInP/GaAs/Ge

En la figura se muestra la estructura típica de este tipo de células. Consiste en una unión p-n de GaAs pasivada con una capa de GaInP. El contacto metálico de la parte superior cubre parcialmente la superficie para que pueda pasar la luz. En la parte inferior, el contacto cubre toda la superficie.

La capa antirreflectante A/R se usa para acoplar los índices de reflexión del semiconductor y del medio para minimizar las pérdidas por reflexión debidas a la diferencia de los índices de refracción entre el medio exterior y el material semiconductor, GaAs.

La capa GaInP se denomina capa ventana. El GaAs tiene una elevada velocidad de recombinación superficial, esto deteriora las prestaciones del dispositivo. La capa ventana soluciona esta situación ya que pasiva la superficie de la unión mediante un material que apenas absorbe luz y que acopla en red.

El espesor y dopado de la unión es diferente según el tipo de polaridad que se elija, n sobre p ó p sobre n. La longitud de difusión de los electrones en GaAs tipo p es de 2 a 10 veces superior a la de huecos en el GaAs tipo n teniendo el mismo valor de dopado. Así como que la movilidad de los electrones en el GaAs es de 10 a 20 veces mayor que la movilidad de los huecos. A la vista de estos valores podemos decir que una configuración n sobre p es lo más óptimo ya que en esta configuración se fuerza a que la fotogeneración se dé mayoritariamente en la zona p y las zonas sensibles a pérdidas resistivas sean de material tipo n.

Las células multiunión son mucho más caras de fabricar que las de silicio, el precio del GaAs es entre 10-15 veces más caro, por lo que a la hora de diseñarlas hay que tener en cuenta el coste. Para ahorrar en coste de materiales se emplean concentradores que permiten reducir el tamaño del semiconductor requerido por vatio generado, es decir si conseguimos concentrar la luz solar x veces, la célula nos proporcionará x veces la potencia eléctrica que generaría sin concentrar.

Se ha conseguido concentrar la energía recibida por la célula en varios miles de soles para células de 1mm^2 .

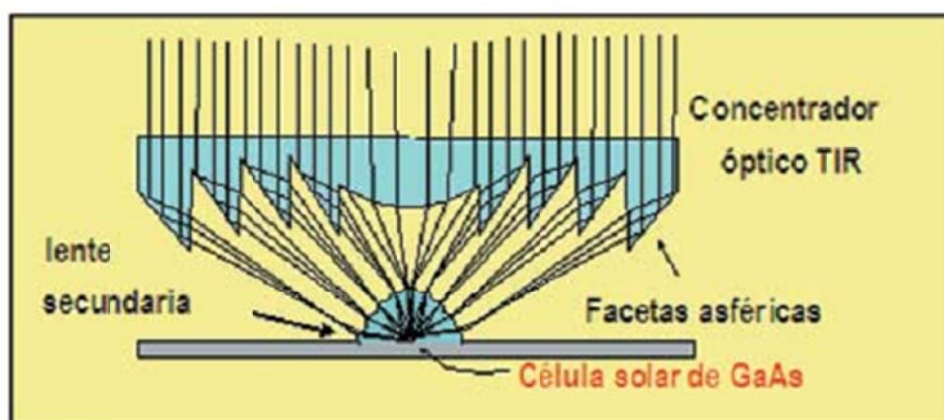


Figura 7: Esquema de un prototipo de concentración fotovoltaica

En la fabricación de este tipo de células, de una oblea pueden sacarse cientos e incluso miles de células, esto dependerá del tamaño que tenga cada célula, que a su vez condicionará el área y la capacidad de concentración del receptor.

Cada célula lleva una mecánica de concentración, que puede tener una sola estructura óptica o dos estructuras ópticas independientes donde la primaria se coloca en la superficie del receptor y la secundaria se sitúa directamente sobre el chip. El factor de concentración solar geométrica es la relación entre el área del receptor, y el área del chip donde se concentra el sol.

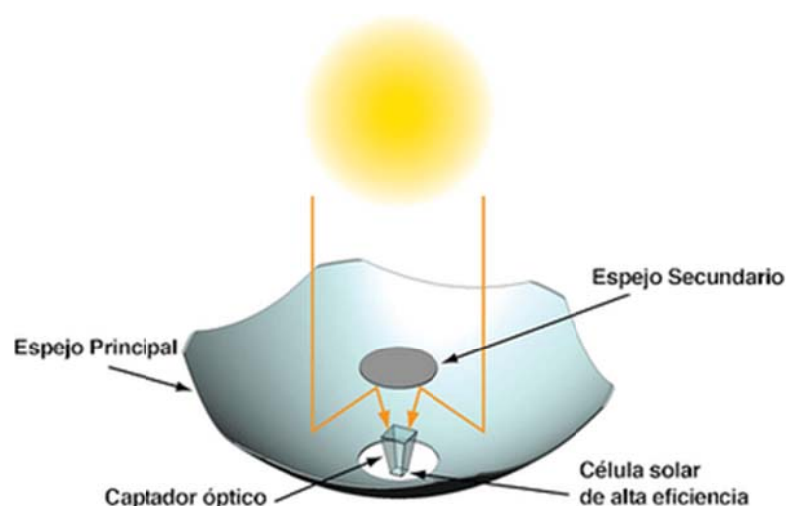


Figura 8 Célula solar con concentrador

El concentrador podría emplearse en cualquier tipo de célula, pero aquí entra el problema de la temperatura, para la que los semiconductores III-V están preparados mucho mejor preparados que el silicio no. Las células de Silicio pierden 2,09 mV por cada grado de aumento de la temperatura, mientras que las de GaAs tan solo un 1,76 mV. En términos generales las células de silicio pierden un 10% de potencia cuando la temperatura aumenta 23°C, mientras que las de GaAs deben aumentar su temperatura en 39°C para perder el mismo porcentaje de potencia.

Concentración ($T = 300$ K)	1X	10X	100X	1000X
Silicio ($dV/dT = \text{mV}/^{\circ}\text{C}$)	2,09	2,00	1,69	1,49
GaAs ($dV/dT = \text{mV}/^{\circ}\text{C}$)	1,76	1,67	1,36	1,16
Si/GaAs ($dV/dT)/(dV/dT)$	1,19	1,20	1,24	1,28

Tabla. II. Coeficientes de variación térmica para el Si y el GaAs para distintas concentraciones.

Además del coste de fabricación, otro de los problemas de estos sistemas es que, para que sean competitivos con los sistemas fotovoltaicos tradicionales, la durabilidad del módulo fotovoltaico de concentración tiene que ser comparable con la de los módulos convencionales, que tienen garantías de 30 años. Por lo tanto es necesario, poder evaluar antes de la industrialización el comportamiento de las células solares de alta concentración durante los 30 años de vida esperada en campo. Para

poder conseguir estos resultados en un tiempo admisible desde un punto de vista industrial se realizan ensayos acelerados. En las células solares de semiconductores III-V aparecen una serie de consideraciones que debemos tener en cuenta a la hora de realizar el análisis de fiabilidad:

- Se emplean nuevos materiales y tecnologías, que no están tan probados como en el caso de las células de Si. Tres uniones en una única célula implica que existen diferentes materiales en la misma estructura lo que implica tensiones que pueden crear defectos.
- Debido a que la energía que no se transforma en electricidad se transforma en calor la temperatura de la célula aumenta, y por lo tanto deberemos usar sistemas específicos para disipar el calor. Este aumento de temperatura también repercute en una disminución de la fiabilidad y de eficiencia de energía producida.
- El subsistema productor de energía es más complejo. El sistema debe estar orientado al sol de forma constante. El subsistema está formado por una célula y un sistema óptico concentrador que debe mantenerse alineado. Las pérdidas de potencia en el sistema será el producto de la pérdida de potencia ocasionadas por cada uno de los subsistemas.

Debido a estas diferencias respecto a sistema solar convencional debemos determinar que tampoco son similares a nivel de fiabilidad, ya que la tecnología empleada es muy diferente tanto en el semiconductor empleado como en la estructura del módulo solar. Por estos motivos es muy importante el estudio de la fiabilidad de los nuevos sistemas solares de concentración.

3 Fiabilidad

3.1. Definición de fiabilidad

El origen de la fiabilidad moderna se remonta a después de la segunda guerra mundial. A partir de la guerra de Corea, en la cual se llevó a cabo el análisis de fallos en equipamiento electrónico militar complejo por parte del departamento de Defensa de EE.UU. y la Industria Electrónica, se constituyó AGREE (Advisory Group on Reliability of Electronic Equipment) en 1952. Fruto de sus trabajos es la norma MILDSTD 781 Reliability Qualification and Production Acceptance Tests y la MIL_STD 785, Reliability Programmes for Systems and Equipment.

Desde entonces hasta la actualidad la fiabilidad ha avanzado de forma importante y en la actualidad todo producto nuevo tiene que pasar por un análisis de fiabilidad.

La Fiabilidad de un producto es la medida de su capacidad para realizar su función cuando es requerida, siempre que dicho dispositivo se emplee en unas condiciones establecidas y durante un periodo establecido, dicho de otro modo; la permanencia de la calidad de los productos o servicios a lo largo del tiempo. Como en términos de fiabilidad no es posible asegurar el funcionamiento de un producto hay que medirlo en términos de probabilidad.

Hay diferencia entre fiabilidad y calidad, la calidad garantiza que el producto sale de fábrica en condiciones óptimas, y la fiabilidad garantiza que el producto permanece en condiciones óptimas durante un determinado tiempo.

Por lo tanto la fiabilidad es una probabilidad. La especificación de un tiempo dado y de unas condiciones de empleo no bastan para conocer con antelación de forma determinística la vida de buen funcionamiento de un dispositivo. Ocurre que un dispositivo nunca es totalmente conocido. Hasta los productos fabricados con máximas garantías en cuanto a materiales, procesos y control de calidad exhiben cierta variación en su capacidad de supervivencia. Aun suponiendo que las condiciones de funcionamiento fueran rigurosamente idénticas, los tiempos de "funcionamiento óptimo" de los dispositivos de un mismo lote de fabricación presentarán una apreciable dispersión.

Por funcionamiento satisfactorio se entiende el cumplimiento de unas actuaciones especificadas. Al cese indebido de este funcionamiento satisfactorio se le llama fallo. Por lo tanto en un estudio de fiabilidad es necesario definir de forma clara

cuando el equipo está en estado de funcionamiento o fallo siendo ambos estados excluyentes. La fiabilidad, según la definición, es función del tiempo o parámetro equivalente. Para ciertos dispositivos, el tiempo no es el parámetro que mejor mide la exposición al fallo, siendo más conveniente emplear como variable independiente el número de ciclos de funcionamiento, por ejemplo.

Naturalmente, la probabilidad de supervivencia de un dispositivo depende de las condiciones de funcionamiento. Si no se fijan éstas, el concepto de fiabilidad carece de sentido. Pero las condiciones de funcionamiento usualmente definidas no suelen cubrir todos los factores capaces de producir fallos. Por ejemplo, sin salirse de los límites especificados para un motor de aviación (régimen, temperaturas, presiones), dicho motor puede ser utilizado de formas muy diferentes (despegues más o menos frecuentes según la longitud de las etapas; elevación diferente de los aeropuertos sobre el nivel del mar; condiciones ambientales variables de temperatura, humedad, atmósfera corrosiva, y otros factores menos evidentes). Por lo tanto, si se pretende reducir la dispersión de los datos estadísticos sobre fallos de un componente o sistema a fin de poder hacer predicciones útiles, se debe especificar con el máximo detalle posible las condiciones ambientales internas y externas (respectivamente, ambiente inducido y ambiente natural). Esto, desgraciadamente, obligará a descartar de la muestra todas las observaciones que no correspondan a estas condiciones de empleo más excluyentes.

Los primeros libros sobre fiabilidad son de la década de los 60 y en la actualidad hay numerosos documentos que cubren los diferentes aspectos de la fiabilidad.

3.2. Fallos y sus clases

Fallo es la incapacitación de un dispositivo para realizar su función dentro de unos límites definidos de actuación. Se suelen dividir en fallos catastróficos y por degradación.

3.2.1. Fallos catastróficos y fallos por degradación

Se suelen llamar fallos catastróficos a los que son súbitos y totales. Son consecuencia de un brusco cambio en un parámetro o característica operativa.

Los fallos por degradación (también llamados no catastróficos o por deriva) son los que obedecen a una degradación progresiva de algún parámetro de actuación que termina por salirse de unas tolerancias o límites de funcionamiento preestablecidos.

En ocasiones el fallo de un componente no provoca el fallo del sistema por diversas razones: la funcionalidad del componente no es crítico, el sistema está diseñado de tal forma que varios componentes realizan la misma función, Un fallo por degradación de un componente puede causar un fallo catastrófico del sistema. En los circuitos complejos, existe la posibilidad de que el sistema falle a causa de una desafortunada combinación de derivas de los parámetros internos, sin poder decir que ningún componente haya fallado.

3.2.2. Modos, mecanismos y causas del fallo

Se llama modo de fallo a la forma en que se manifiesta el fallo. Por ejemplo, el sistema fotovoltaico con células solares puede presentar distintos tipos de fallos:

- Degradación óptica del concentrador.
- Corrosión de la célula
- Soldadura
- Cortocircuito
- Circuito abierto.

Los tres primeros son modos de degradación y los dos últimos, modos catastróficos.

Por mecanismo de fallo se entiende algo más interno. Un mismo modo de fallo puede presentarse por varios mecanismos. Finalmente, cada mecanismo de fallo obedecerá a una causa o a una combinación de causas.

1.1.1. Tipos de fallo

Los fallos se pueden clasificar en función del periodo de vida en que se encuentren los dispositivos. Las tres tipos de fallos son los infantiles, los aleatorios (por azar) y los debidos al desgaste.

Los fallos infantiles son los debidos a cierto defecto de la pieza o conjunto como resultado de una deficiencia de diseño, fabricación o inspección y suelen

aparcer al inicio de la vida del dispositivo. Se ejerce una acción correctiva, se eliminan las causas de fallo y se hacen ensayos de verificación. Sin embargo, en ciertos casos no es posible eliminar económicamente mediante diseño de algunos mecanismos de fallo de una pieza. Es globalmente menos costoso producir muchas piezas y separar las unidades potencialmente defectuosas mediante ensayos de purga (burn-in), muy común en electrónica.

Los sistemas se suelen someter a un burn-in haciéndolos funcionar durante cierto periodo antes de la entrega para asegurarse de la detección y eliminación de posibles fallos infantiles. Claro que no es aplicable a todos los casos, si hablamos de misiles y satélites se consideran no reparables durante su uso, pero si es aplicable para sistemas comerciales como radares.

El segundo tipo de fallo, son los fallos por azar o aleatorios, los cuales son averías accidentales, debidas a un pico de sollicitación capaz de afectar a cualquier componente normal o subnormal, nuevo o usado.

Y por último, los fallos por desgaste, en los cuales los componentes han perdido resistencia al fallo por el uso.

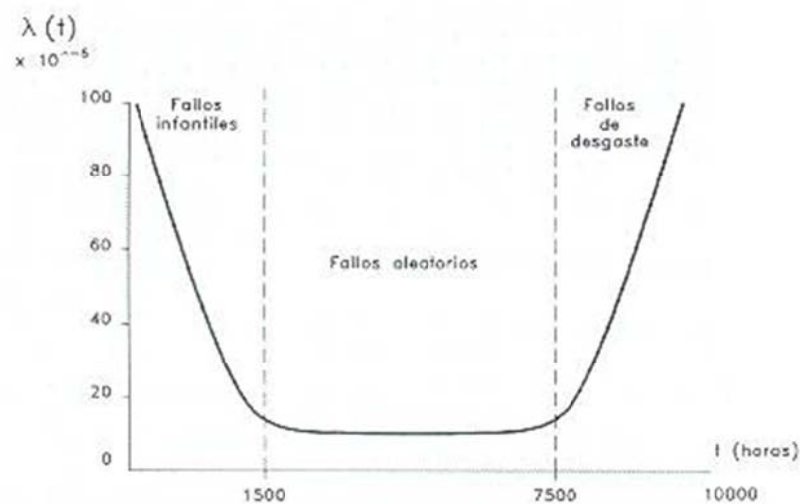


Figura 9: Distribución de fallos a lo largo del tiempo

1.1.2. Fallos primarios y secundarios.

Además de la anterior clasificación, se pueden distinguir fallos primarios o secundarios, siendo el primario el fallo de un componente cuando no se debe al fallo

de otro u otros componentes. Y los fallos inducidos por el fallo de otro u otros componentes se llaman secundarios o dependientes.

3.3. La fiabilidad como función del tiempo

3.3.1. Introducción

Desde el punto de vista matemático, la función de fiabilidad de un producto es la probabilidad de que un sistema falle de 0 a t .

3.3.2. Fiabilidad e in fiabilidad

Para crear un modelo matemático para la probabilidad de fallo, consideramos el funcionamiento de un determinado elemento en el medio para él especificado. Definimos la variable aleatoria como el tiempo durante el que el elemento funciona satisfactoriamente antes de que se produzca un fallo. La probabilidad de que el elemento proporcione unos resultados satisfactorios en el momento t se puede definir como fiabilidad. La designamos $R(t)$ y si el tiempo de fallo es T entonces podremos escribir:

$$R(t) = \Pr(T > t) \quad (6)$$

De una forma práctica si designamos:

$N_s(t)$ = N° de elementos en funcionamiento en el instante t

$N(0)$ = N° de elementos en funcionamiento al principio

$N_f(t)$ = N° de elementos averiados hasta el momento t

se cumplirá:

$$N(0) = N_f(t) + N_s(t) \quad (7)$$

$$R(t) = \frac{N_s(t)}{N(0)} = 1 - \frac{N_f(t)}{N(0)} \quad (8)$$

La fiabilidad $R(t)$ está relacionada con la función complementaria llamada in fiabilidad $F(t)$. $F(t)$ es la probabilidad de que el dispositivo no falle después del instante t es decir, de que $\tau \leq t$.

Por lo tanto la in fiabilidad valdrá:

$$F(t) = \frac{N_f(t)}{N(0)} \quad (9)$$

Cumpléndose que:

$$F(t) = 1 - R(t) \quad (10)$$

3.3.3. Distribución de fallos

Se parte de la hipótesis de que el tiempo hasta el fallo de un dispositivo es una variable aleatoria, a la que llamaremos τ , que puede tomar cualquier valor real t desde 0 a ∞ . Si $F(t)$ es la función de distribución y $f(t)$ la función de densidad de τ , se tiene:

$$F(t) = P(\tau \leq t) = \int_0^t f(x) dx \quad (11)$$

$$F(0) = 0, F(\infty) = 1 \quad (12)$$

La probabilidad complementaria de ésta, es decir, la probabilidad de que el fallo se produzca después de t es la función de fiabilidad:

$$R(t) = 1 - F(t) = P(\tau > t) = \int_t^{\infty} f(x) dx \quad (13)$$

$$R(0) = 1, R(\infty) = 0 \quad (14)$$

3.3.4. Función densidad de probabilidad de fallos

La función densidad de probabilidad de fallos es la probabilidad de que un dispositivo cualquiera tenga un fallo entre los instantes t y $t + dt$. Se la denomina $f(t)$ y matemáticamente tiene la expresión:

$$f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \quad (15)$$

Relación entre $f(t)$, $\lambda(t)$ y $R(t)$

Se cumple que la probabilidad de producirse una avería en un elemento entre t y $t + dt$ o sea $f(t) dt$ es igual a la probabilidad de que funcione hasta t (fiabilidad) por la probabilidad de que falle entre t y $t + dt$. Puesto de forma matemática se cumplirá:

$$f(t) dt = R(t) \cdot \lambda(t) dt \quad (16)$$

En la figura se puede ver la representación gráfica de los parámetros expuestos para un caso general.

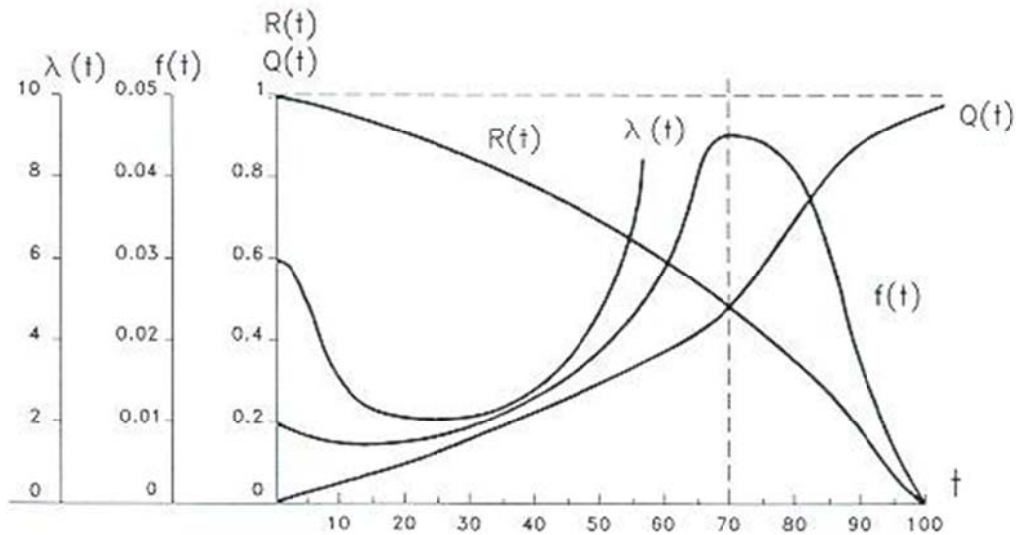


Figura 10: Representación gráfica general de los parámetros de fiabilidad siendo $Q(t)=F(t)$

3.3.5. Vida media

Entre los parámetros de la distribución de fallos de un dispositivo, destaca por su importancia la esperanza matemática (momento de primer orden respecto del origen) de τ :

$$\theta = E(\tau) = \int_0^{\infty} t f(t) dt \quad (17)$$

La esperanza matemática del tiempo hasta el fallo se llama vida media o tiempo medio hasta el fallo (*mean time to failure* o MTTF= θ).

3.3.6. Tasa de fallo

En una población de dispositivos nominalmente idénticos, la proporción esperada de supervivientes en el instante t será $R(t)$. La probabilidad de que falle un dispositivo superviviente en el intervalo $(t, t + \Delta t)$ es una probabilidad condicional que se expresa así:

$$P(t < \tau \leq t + \Delta t | \tau > t) = \frac{P(t < \tau \leq t + \Delta t)}{P(\tau > t)} = \frac{R(t) - R(t + \Delta t)}{R(t)} \quad (18)$$

Este valor, dividido por la longitud Δt del intervalo, se llama tasa media de fallo en $(t, t + \Delta t)$:

$$z(t, \Delta t) = \frac{R(t) - R(t + \Delta t)}{\Delta t R(t)} \quad (19)$$

El límite de $z(t, \Delta t)$ cuando Δt tiende a cero es la tasa instantánea de fallo o simplemente tasa de fallo en t :

$$z(t) = -\frac{1}{R(t)} \frac{dR(t)}{dt} \quad (20)$$

La expresión anterior pone de manifiesto que la tasa de fallo es una medida de la variación de la fiabilidad en el tiempo. La tasa de fallo es la densidad de probabilidad de fallo en t condicionada a no haber fallado antes de t .

La tasa de fallos se medirá en fallos por año, por hora, por millón de horas o por 10^9 horas (FIT).

3.3.7. La vida media en función de la fiabilidad

En determinadas condiciones, conviene calcular la vida media a partir de $R(t)$. Para ello se integra por partes:

$$\theta = \int_0^{\infty} t f(t) dt = - \int_0^{\infty} t dR(t) = \left[tR(t) \right]_0^{\infty} + \int_0^{\infty} R(t) dt \quad (21)$$

Puesto que para $t = 0$, $tR(t) = 0$, el primer sumando valdrá:

$$- \lim_{t \rightarrow \infty} [tR(t)] \quad (22)$$

Límite indeterminado al que aplicamos la regla de L'Hôpital:

$$\lim_{t \rightarrow \infty} [tR(t)] = \lim_{t \rightarrow \infty} \frac{t}{R^{-1}(t)} = \lim_{t \rightarrow \infty} \frac{1}{R^{-2}(t) f(t)} = \lim_{t \rightarrow \infty} \frac{R(t)}{z(t)} \quad (23)$$

Este límite es nulo siempre que $z(t)$ no tienda a cero, o sea un infinitésimo de orden inferior a $R(t)$. Como en los casos usuales $z(t)$ no es decreciente, esto se verifica. Por tanto, podemos escribir:

$$\theta = \int_0^{\infty} R(t) dt \quad (24)$$

3.3.8. Representación gráfica

La siguiente figura representa una curva de densidad. Se ha trazado con datos de una tabla de mortalidad humana (t en años), pero no difiere mucho en tendencia general de la densidad de fallos de algunos componentes electrónicos (excepto la unidad de tiempo). Las ordenadas decrecientes cerca del origen son imputables a la mortalidad infantil, que es un hecho real en dichos componentes. Tras un período de densidad relativamente baja, la mayor parte de los dispositivos fallan por desgaste.

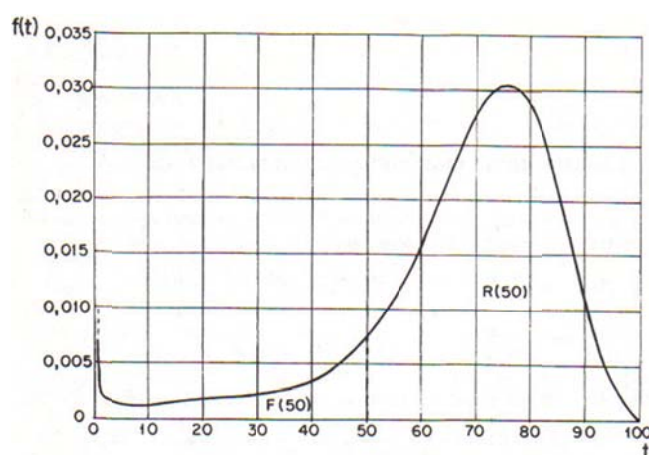


Figura 11: Función de densidad de fallo

La figura siguiente muestra las curvas correspondientes a la función de distribución (infiabilidad) $F(t)$ y a la función de fiabilidad $R(t)$. Se han trazado por integración de $f(t)$. Por ejemplo, la ordenada $F(50)$ mide el área bajo la curva de densidad desde el origen hasta $t = 50$. Su valor, aproximadamente 0,12, indica que la probabilidad de que un dispositivo de este tipo hipotético falle antes de o en el instante 50 es 0,12 (el 12 %). La ordenada $R(50)$ mide el área bajo la curva de densidad desde $t = 50$ hasta el infinito. Como era de esperar, vale $1 - F(50) = 1 - 0,12 = 0,88$ (el 88 %). Es la probabilidad de que un dispositivo sobreviva al menos hasta el instante 50. La curva $R(t)$, permite hallar la vida media de este tipo de dispositivo. Midiendo el área por ella encerrada, se ve que es de unas 68 unidades de tiempo (68 años para la población humana a la que correspondían las tablas de mortalidad).

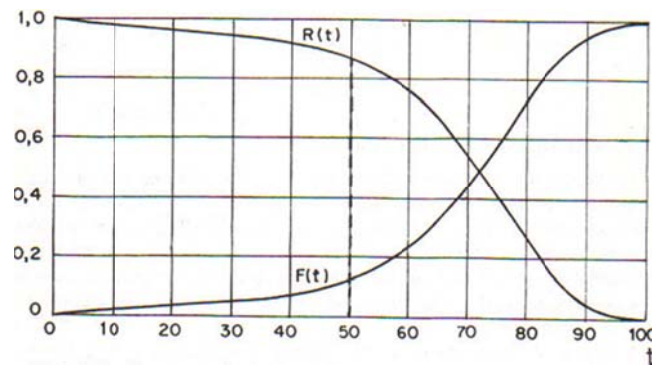


Figura 12: Curva in fiabilidad $F(t)$ y Fiabilidad $R(t)$

La figura representa la función de tasa de fallo de la distribución considerada. Su tendencia general no es del todo diferente a la de la densidad, con la diferencia de que no decrece al final. La explicación es sencilla. La densidad de fallo decrece porque cada vez quedan menos supervivientes que puedan fallar. Como la tasa de fallo está referida en cada instante al número de supervivientes, no resulta afectada por los fallos anteriores. El que $f(50)$ valga 0,0072 indica que la frecuencia de fallo en el instante $t = 50$ es de un 0,72 por 100 de la población inicial por unidad de tiempo. Como en ese instante el número de supervivientes es el 88 por 100 de la población inicial:

$$z(50) = \frac{f(50)}{R(50)} = \frac{0,0072}{0,88} = 0,0082 \quad (25)$$

que indica que la tasa instantánea de fallo es del 0,82 por 100 de los supervivientes parra $t= 50$.

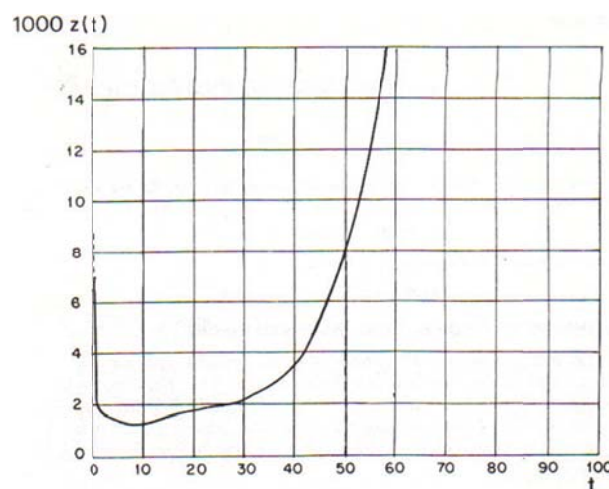


Figura 13: Función de tasa de fallo

3.4. Funciones de distribución estadísticas de fallos

3.4.1. Introducción

Se han propuesto numerosas distribuciones teóricas para describir matemáticamente las leyes de fallo de diversos dispositivos. Las funciones de distribución más utilizadas son la exponenciales, la distribución normal de Weibull, gamma, log-norma, etc., cada una con sus virtudes y su campo de aplicación. Evidentemente, lo correcto es contrastar las *hipótesis de distribución* antes de aceptarlas como razonables.

A continuación se explican varias distribuciones comúnmente utilizadas.

3.4.2. La curva de bañera

Anteriormente se han descrito los tipos de fallos, según los cuales, a semejanza con los seres humanos, los componentes y, en general, los dispositivos no reparables, mueren por taras infantiles, por azar o por desgaste.

La curva de tasa de fallo correspondiente tendrá una forma como la que muestra la figura curva de bañera, que hace que los americanos la denominen *bathtub curve* (curva de bañera). Nótese que la curva de fuerza de mortalidad humana es bastante parecida.

La curva de bañera revela tres partes bien definidas:

- Fallos infantiles: esta etapa se caracteriza por tener una elevada tasa de fallos que desciende rápidamente con el tiempo. Estos fallos pueden deberse a diferentes razones como equipos defectuosos, instalaciones incorrectas, errores de diseño del equipo, desconocimiento del equipo por parte de los operarios o desconocimiento del procedimiento adecuado.
- Fallos normales: etapa con una tasa de errores menor y constante. Los fallos no se producen debido a causas inherentes al equipo, sino por causas aleatorias externas. Estas causas pueden ser accidentes fortuitos, mala operación, condiciones inadecuadas y otros sucesos fortuitos.

- Fallos de desgaste: etapa caracterizada por una tasa de errores rápidamente creciente. Los fallos se producen por desgaste natural del equipo debido al transcurso del tiempo.

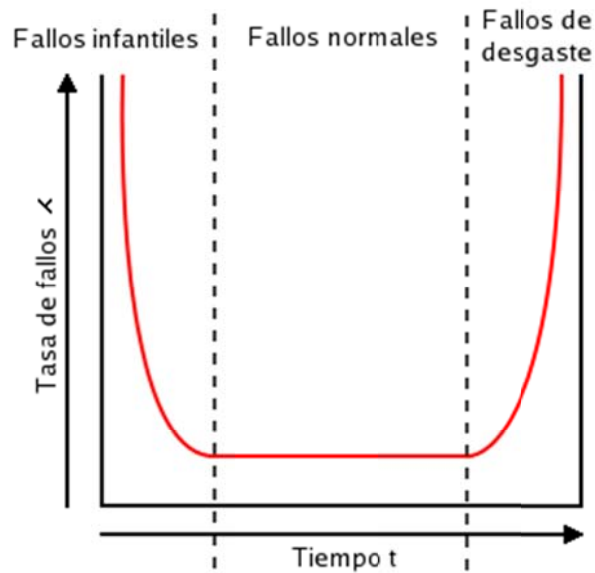


Figura 14: Curva de bañera

No se conoce ninguna distribución analíticamente expresable en forma sencilla que dé esta forma para la curva $z(t)$. Se ha propuesto el empleo de una función de densidad de la forma:

$$f(t) = af_1(t) + bf_2(t) + cf_3(t) \quad (26)$$

en que $f_1(t)$, $f_2(t)$ y $f_3(t)$ son funciones de densidad que describen cada una uno de los períodos señalados. Los coeficientes a , b y c son, respectivamente, las probabilidades de que el fallo sea infantil, por azar y por desgaste, por lo que $a + b + c = 1$. Con ello se cumple también la condición:

$$\int_0^{\infty} f(t) dt = 1 \quad (27)$$

Para describir el período infantil hay que emplear una distribución con tasa de fallo decreciente. Determinados casos de las distribuciones gamma y de Weibull se ajustan a esta condición. Se usa, por ejemplo, la densidad de Weibull:

$$f_1(t) = \frac{1}{100} \left(\frac{1}{50} \right)^{-0.5} e^{-\left(\frac{1}{50} \right)^{0.5}} \quad (28)$$

Para la vida útil, con tasa de fallo constante, es forzoso emplear una densidad exponencial, como la siguiente:

$$f_2(t) = \frac{1}{100000} e^{-t/100000} \quad (29)$$

Finalmente, el período de desgaste, caracterizado por una tasa de fallo creciente, exige una densidad en forma de campana, como son las normales y ciertos casos de la de Weibull. Se emplea la densidad normal:

$$f_3 = \frac{1}{1000\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-10000}{1000}\right)^2} \quad (30)$$

3.4.3. Período infantil

Se han propuesto diversas maneras de tratar analíticamente el período infantil. Un enfoque consiste en suponer que la población inicial incluye una proporción de dispositivos subnormales con tasa de fallo constante superior a la de los dispositivos normales.

Es frecuente en electrónica someter a los componentes, antes de montarlos en los circuitos, a un período de funcionamiento de, por ejemplo, 180 a 200 horas para eliminar los subnormales. Este proceso, realizado a menudo por el fabricante del componente con lotes de fabricación completos, se llama burn-in. Se llamara al burn-in, purga.

Otra manera de eliminar componentes con taras infantiles es hacer funcionar el sistema del que forman parte. Esta operación, llamada debugging (supresión de chinches), es la que hace el que compra un automóvil, haciendo votos porque todos los componentes subnormales fallen en el período de garantía. Es frecuente que las condiciones de recepción de un producto incluyan un número dado de horas de ensayo previo, pero esto no es siempre aplicable. En estos casos, se purgan los componentes antes de montarlos.

3.4.4. La distribución exponencial

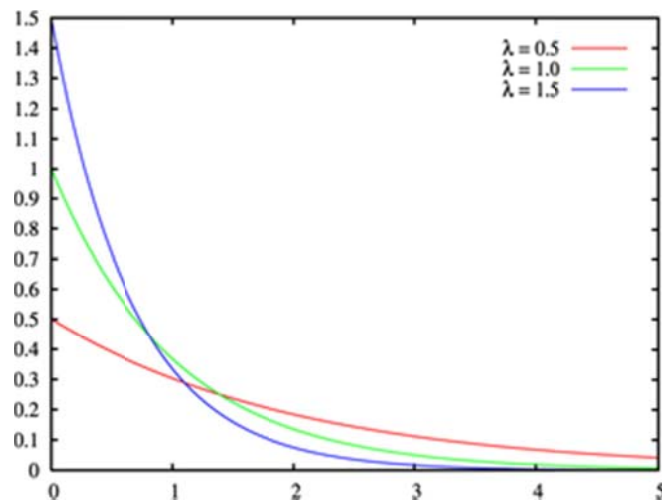


Figura 15: Distribución exponencial

La distribución exponencial es un caso especial de la distribución gamma, ambas tienen un gran número de aplicaciones. Las distribuciones exponencial y gamma juegan un papel importante tanto en teoría de colas como en problemas de confiabilidad. El tiempo entre las llegadas en las instalaciones de servicio y el tiempo de fallo de los componentes y sistemas eléctricos, frecuentemente involucran la distribución exponencial. La relación entre la gamma y la exponencial permite que la distribución gamma se utilice en tipos similares de problemas.

La variable aleatoria x tiene una distribución exponencial, con parámetro b , si su función de densidad es:

$$f(x) = \frac{1}{b} e^{-\frac{x}{b}}, \quad x > 0 \quad (31)$$

$f(x) = 0$ en cualquier otro caso

donde $b > 0$

La media y la variancia de la distribución exponencial son:

$$m = b \quad \text{y} \quad s^2 = b^2$$

Las aplicaciones más importantes de la distribución exponencial son aquellas situaciones en donde se aplica el proceso de Poisson, es necesario recordar que un proceso de Poisson permite el uso de la distribución de Poisson. Recuérdese

también que la distribución de Poisson se utiliza para calcular la probabilidad de números específicos de “eventos” durante un período o espacio particular. En muchas aplicaciones, el período o la cantidad de espacio es la variable aleatoria. Por ejemplo un ingeniero industrial puede interesarse en el tiempo T entre llegadas en una intersección congestionada durante la hora de salida de trabajo en una gran ciudad. Una llegada representa el evento de Poisson.

La relación entre la distribución exponencial (con frecuencia llamada exponencial negativa) y el proceso llamado de Poisson es bastante simple. La distribución de Poisson se desarrolló como una distribución de un solo parámetro λ , donde λ puede interpretarse como el número promedio de eventos por unidad de “tiempo”. Considérese ahora la variable aleatoria descrita por el tiempo que se requiere para que ocurra el primer evento. Mediante la distribución de Poisson, se encuentra que la probabilidad de que no ocurran en el espacio hasta el tiempo t está dada por:

$$p(0, \lambda t) = \frac{E^{-\lambda t} (\lambda t)^0}{0!} ; E=2.718 \quad (32)$$

Ahora puede utilizarse lo anterior y hacer que X sea el tiempo para el primer evento de Poisson. La probabilidad de que el período hasta que ocurre el primer evento de Poisson exceda x es la misma que la probabilidad de que no ocurra un evento de Poisson en x . Esto último por supuesto está dado por $E^{-\lambda t}$. Como resultado,

$$P(X \geq x) = E^{-\lambda t} \quad (33)$$

Entonces, la función de distribución es:

$$F(t) = 1 - e^{-\lambda t} \quad (34)$$

La función de densidad de la distribución exponencial:

$$f(t) = \lambda e^{-\lambda t} \quad (35)$$

Su función de fiabilidad

$$R(t) = e^{-\lambda t} \quad (36)$$

Y su tasa de fallo

$$z(t) = \lambda = \frac{1}{\beta} \quad (37)$$

Nótese que la media de la distribución exponencial es el parámetro β , el recíproco del parámetro en la distribución de Poisson. Con frecuencia se dice que la distribución de Poisson no tiene memoria, lo cual implica que las ocurrencias en períodos de tiempo sucesivos son independientes. Aquí el parámetro importante β es el tiempo promedio entre eventos. En teoría de la confiabilidad, donde la falla de un equipo concuerda con el proceso de Poisson, β recibe el nombre de tiempo promedio entre fallas. Muchas descomposturas de equipo siguen el proceso de Poisson, y entonces la distribución exponencial es aplicable.

Se aplica la distribución exponencial en componentes purgados de vida útil muy larga, que excede de la vida de servicio de los sistemas de que forman parte, en componentes purgados que se sustituyen preventivamente antes de que llegue el desgaste, en sistemas en serie compuestos de bloques exponenciales y, en la práctica, en sistemas reparables muy complejos en que no haya redundancias dominantes.

Es evidente que en los dos primeros casos la vida media θ no lo es en rigor, pues no se tiene en cuenta más que la parte central de la curva de bañera. Utilizando el modelo descrito en el punto la curva de bañera, es fácil ver que la densidad $f_3(t)$ desplaza a la izquierda la vida media. Por ello, el parámetro θ , tratándose de componentes sujetos a desgaste, aunque empleados sólo durante su vida útil, es una "vida media ficticia".

$$\theta = \int_0^{\infty} R(t)dt = \int_0^{\infty} e^{-\lambda t} dt = \frac{1}{\lambda} \quad (38)$$

En el caso de sistemas que fallan exponencialmente y son reparables, θ es el *tiempo medio entre fallos* (*mean time between failures* o MTBF). Evidentemente, los sistemas reparables tienen una nueva vida después de cada reparación. La distribución de fallos, en general, puede variar a cada fallo, por desgaste de los componentes que no han fallado y siguen en el sistema. No obstante, para los sistemas en serie como rigurosa o aproximadamente exponenciales, las sucesivas distribuciones son aproximadamente iguales, conservando el mismo valor de θ . Aunque el término MTBF se ha empleado para casos no exponenciales y también para componentes sencillos, ahora se aplicará sólo al tiempo medio entre fallos de sistemas exponenciales.

3.4.5. La distribución normal

La función que nos define esta distribución es:

$$f(x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (39)$$

Al dar a la función los valores de μ , σ^2 y valores a x , obtendremos la función de distribución en cuestión, la que tiene forma de campana, por lo que también se le conoce como campana de Gauss. Hay un número infinito de funciones de densidad Normal, una para cada combinación de μ y σ . La media μ mide la ubicación de la distribución y la desviación estándar σ mide su dispersión.

- Es simétrica con respecto a su eje vertical.
- Es asintótica con respecto a su eje horizontal; esto quiere decir que jamás va a tocar el eje de las equis.
- El área total bajo la curva es 1.
- Se observa que el 68.26% de los datos se encuentran en $\mu \pm \sigma$, se observará que aproximadamente el 68.26% de los datos se encuentran bajo la curva, el 95.44% alrededor de $\mu \pm 2\sigma$, de los datos estará entre esos límites y el 99.74% de los datos dentro de esos límites de $\mu \pm 3\sigma$. Esta característica es a la vez una forma empírica y rápida de demostrar si los datos que se analizan tienen una distribución Normal; ya que para trabajar los datos con esta distribución, debe verificarse que efectivamente así se distribuyen, ya que de no hacerlo, las decisiones que en un momento dado se tomarán de un análisis de los datos con la distribución Normal, serían erróneas.

¿Cómo se determinan probabilidades con la distribución Normal?

Lo más lógico es que la función $f(x, \mu, \sigma^2)$, se integre entre los límites de la variable x ; esto es,

$$p(a \leq x < b) = \int_a^b f(x, \mu, \sigma^2) dx \quad (40)$$

La integral anterior nos daría el área bajo la curva de la función, desde a hasta b , que corresponde o es igual a la probabilidad buscada.

Debido a la dificultad que se presenta para integrar esta función cada vez que sea necesario, lo que se hace es tipificar el valor de la variable x , esto es, x se transforma en un valor de z , de la siguiente manera:

$$z = \frac{x - \mu}{\sigma} = \text{valor} \quad (41)$$

Este valor de z es buscado en una tabla donde vienen áreas asociadas a este valor, y haciendo uso de los valores tabulados, se determina la probabilidad requerida. La tabla que es usada para calcular las probabilidades es la que nos da el área que se muestra a continuación:

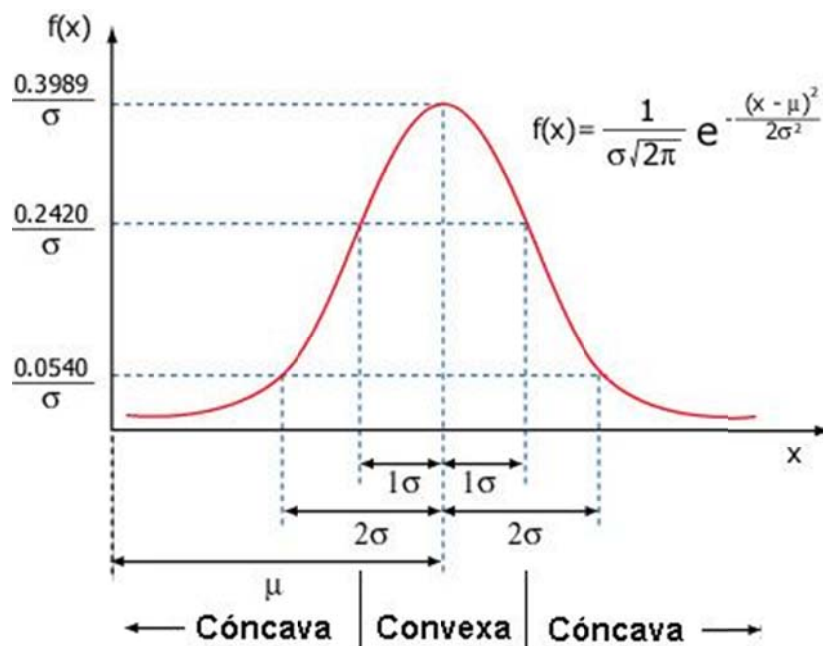


Figura 16: Distribución normal

Particularmente en el caso de un tipo de dispositivo que tenga un solo mecanismo de fallo por desgaste claramente dominante, la distribución normal o de Gauss es un modelo razonable.

En general, los componentes mecánicos y electromecánicos, como engranajes, cojinetes, motores, lámparas, relés etc., fallan predominantemente por desgaste, si bien pueden tener más de un mecanismo de desgaste. En este caso, la densidad puede perder su simetría, siendo conveniente utilizar una distribución de Weibull.

La distribución normal tiene dos parámetros, la media θ y la desviación típica σ (o la varianza σ^2).

Las funciones de densidad, distribución (infiabilidad) y fiabilidad son:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\theta}{\sigma}\right)^2} \quad (42)$$

$$F(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{1}{2}\left(\frac{x-\theta}{\sigma}\right)^2} dx \quad (43)$$

$$R(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_t^{\infty} e^{-\frac{1}{2}\left(\frac{x-\theta}{\sigma}\right)^2} dx \quad (44)$$

La vida media será:

$$E(\tau) = \int_{-\infty}^{\infty} tf(t)dt = \theta \quad (45)$$

La varianza es:

$$V(\tau) = \int_{-\infty}^{\infty} (t - \theta)^2 f(t)dt = \sigma^2 \quad (46)$$

y la desviación típica $D(t)$ es la raíz cuadrada de la varianza, es decir, σ .

La función de tasa de fallo es siempre creciente:

$$z(t) = \frac{e^{-\frac{1}{2}\left(\frac{t-\theta}{\sigma}\right)^2}}{\int_t^{\infty} e^{-\frac{1}{2}\left(\frac{x-\theta}{\sigma}\right)^2} dx} \quad (47)$$

El límite inferior de integración es $-\infty$ en vez de 0. Esto se debe a que la densidad normal no es nula en todo el eje real, y no sólo en el semieje positivo. De esta forma, es posible que el fallo se produzca antes del instante inicial $t = 0$. Este defecto de la distribución normal como modelo de distribución de fallos no es grave si

θ y σ son tales que la cola a la izquierda de $t = 0$ tiene un área despreciable. Para ello, es necesario que θ sea mayor que 3 ó 4 veces σ , que es lo habitual.

La distribución normal típica (media 0 y desviación típica 1) está tabulada y sirve para manejar cualquier distribución normal, que puede tipificarse mediante el cambio de variable:

$$z = \frac{t - \theta}{\sigma} \quad (48)$$

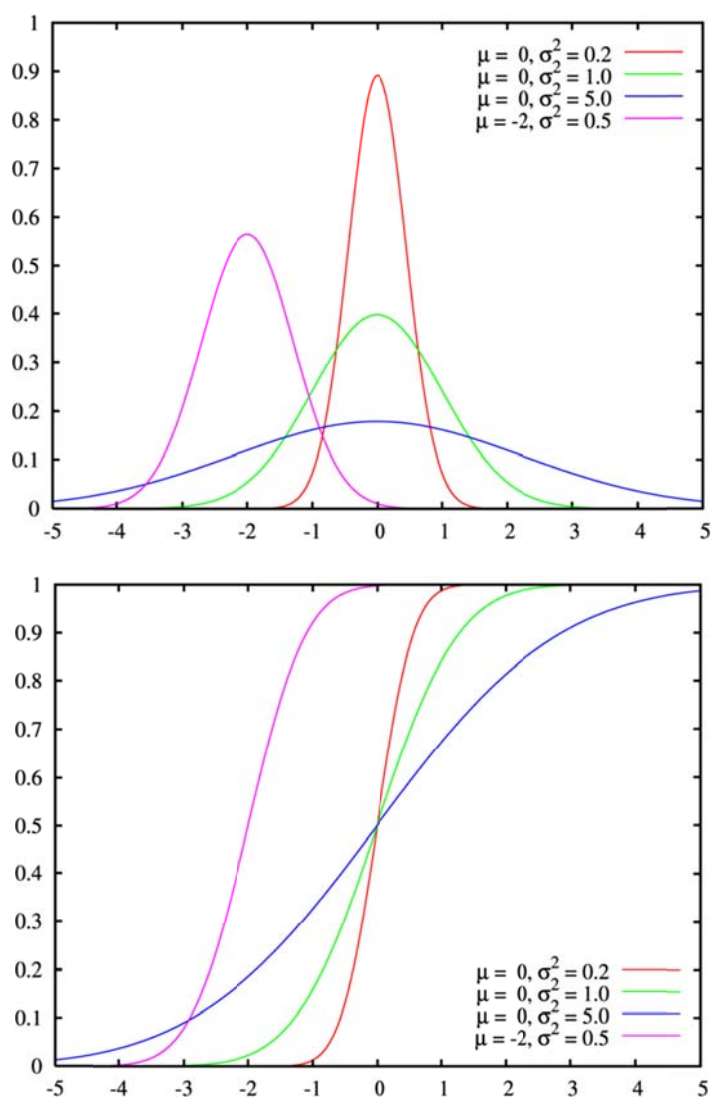


Figura 17 Curvas de densidad de probabilidad, distribución de probabilidad

3.4.6. La distribución log-normal

La distribución lognormal se caracteriza por las siguientes propiedades:

- Asigna a valores de la variable < 0 la probabilidad 0 y de este modo se ajusta a las tasas y probabilidades de fallo que de esta forma sólo pueden ser positivas.
- Como depende de dos parámetros, según veremos, se ajusta bien a un gran número de distribuciones empíricas.
- Es idónea para parámetros que son a su vez producto de numerosas cantidades aleatorias (múltiples efectos que influyen sobre la fiabilidad de un componente).
- La esperanza matemática o media en la distribución lognormal es mayor que su mediana. De este modo da más importancia a los valores grandes de las tasas de fallo que una distribución normal con los mismos porcentajes del 5% y 50% tendiendo, por tanto, a ser pesimista. Esta propiedad se puede apreciar en la figura siguiente.

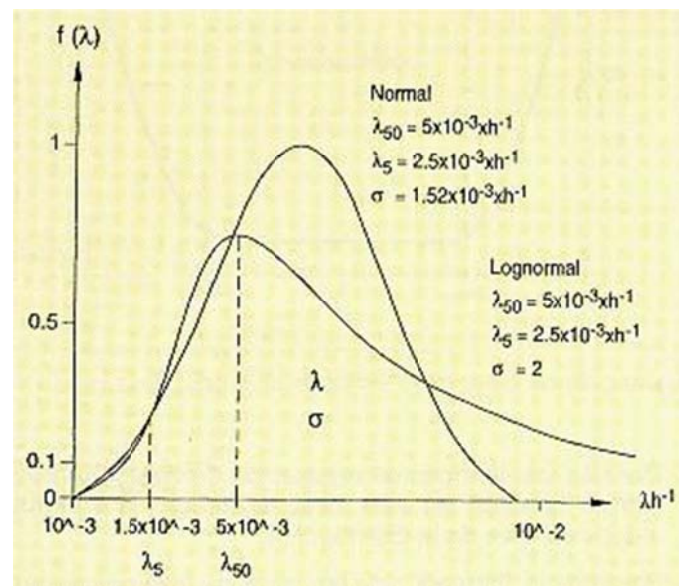


Figura 18: Comparación entre una distribución normal y una lognormal con los mismos percentiles del 5% y 50%. (Distribución normal normalizada a 1, distribución lognormal con el mismo factor)

La distribución lognormal se ajusta a ciertos tipos de fallos (fatiga de componentes metálicos), vida de los aislamientos eléctricos, procesos continuos (procesos técnicos) y datos de reparación y puede ser una buena representación de la distribución de los tiempos de reparación. Es también una distribución importante en la valoración de sistemas con reparación.

La distribución lognormal es importante en la representación de fenómenos de efectos proporcionales, tales como aquellos en los que un cambio en la variable en cualquier punto de un proceso es una proporción aleatoria del valor previo de la variable. Algunos fallos en el programa de mantenimiento entran en esta categoría.

Para evitar el defecto de la distribución normal como distribución de tiempos hasta el fallo, es decir, la no nulidad de su densidad para valores negativos de t , se ha sugerido para el estudio de los fallos por desgaste la distribución log-normal, cuya función de densidad es:

$$f(t) = \frac{1}{t\sigma'\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln t - \theta'}{\sigma'}\right)^2} \quad (49)$$

La densidad log-normal sólo tiene valores reales para $t \geq 0$. Por supuesto, su media y su varianza no son θ' y σ' , sino:

$$\begin{aligned} \theta &= E(\tau) = e^{\theta' + \frac{\sigma'^2}{2}} \\ \sigma^2 &= V(\tau) = e^{2\theta' + \sigma'^2} (e^{\sigma'^2} - 1) \end{aligned} \quad (50)$$

La función de tasa de fallo es inicialmente creciente.

Quizá donde ha encontrado más aceptación la distribución log-normal es en estudios de reparabilidad, como distribución de tiempos de reparación. Dichos tiempos, en efecto, suelen acumularse alrededor de una media, pero algunas reparaciones se prologan de una forma que queda bien representada por la cola derecha de la log-normal.

Se puede resaltar que el parámetro de localización de la distribución normal original se ha convertido ahora en un parámetro de escala coincidente con la media. La dispersión s es también un parámetro de forma según se puede ver en la figura siguiente, donde aparece la función densidad de probabilidad de la distribución lognormal para distintos valores de s (0,4; 0,6; 1,0 y 1,4).

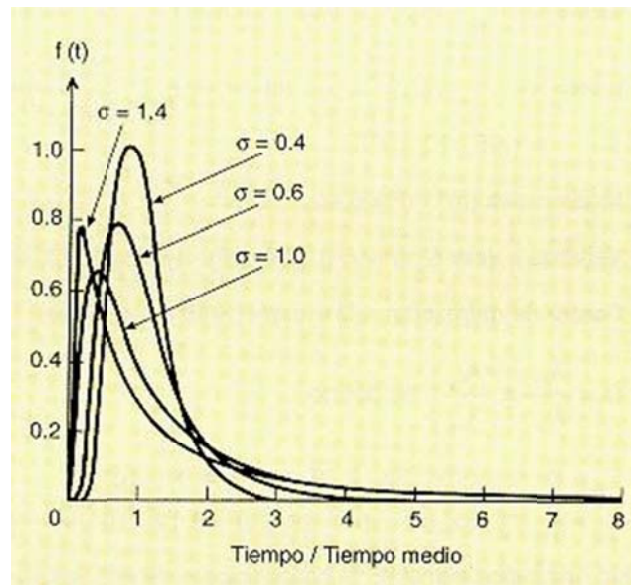


Figura 19: Función densidad de probabilidad para distintos valores de s (0,4 a 1,4)

Para valores medios y altos de s , la distribución lognormal es significativamente asimétrica, pero a medida que s decrece la distribución es más simétrica. Si s se acerca a la unidad, la distribución lognormal es equivalente aproximadamente a la distribución exponencial negativa. También se puede observar que para valores de $s < 0,2$ la distribución lognormal se aproxima a la distribución normal.

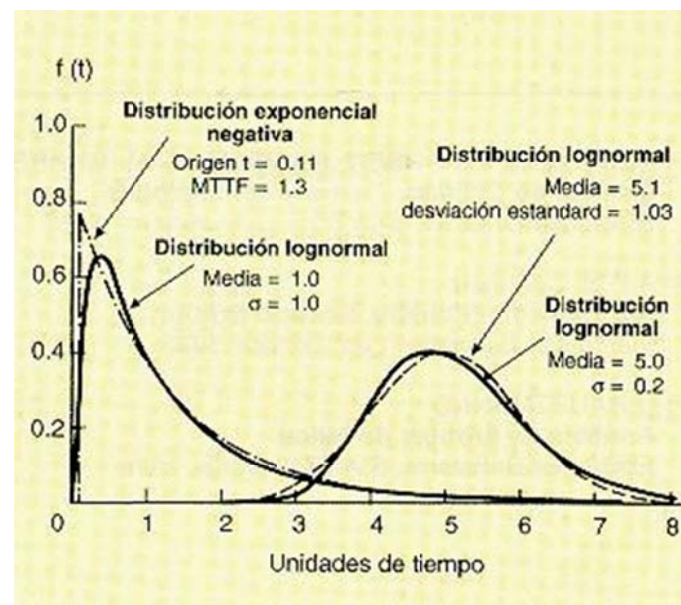


Figura 20: Comparación entre la distribución lognormal y las distribuciones normal y exponencial

3.4.7. La distribución de Weibull

El investigador sueco W. Weibull estudió (1939) la vida de fatiga de los metales y propuso la distribución que lleva su nombre. La distribución de Weibull es hoy de muy amplia aplicación en fiabilidad, por su flexibilidad y por su facilidad de manejo mediante papel probabilístico.

La distribución de Weibull complementa a la distribución exponencial y a la normal, que son casos particulares de aquella, como veremos. A causa de su mayor complejidad sólo se usa cuando se sabe de antemano que una de ellas es la que mejor describe la distribución de fallos o cuando se han producido muchos fallos (al menos 10) y los tiempos correspondientes no se ajustan a una distribución más simple. En general es de gran aplicación en el campo de la mecánica.

Aunque existen dos tipos de soluciones analíticas de la distribución de Weibull (método de los momentos y método de máxima verosimilitud), ninguno de los dos se suele aplicar por su complejidad. En su lugar se utiliza la resolución gráfica a base de determinar un parámetro de origen (t_0). Un papel especial para gráficos, llamado papel de Weibull, hace esto posible. El procedimiento gráfico, aunque exige varios pasos y una o dos iteraciones, es relativamente directo y requiere, a lo sumo, álgebra sencilla.

La distribución de Weibull nos permite estudiar cuál es la distribución de fallos de un componente clave de seguridad que pretendemos controlar y que a través de nuestro registro de fallos observamos que éstos varían a lo largo del tiempo y dentro de lo que se considera tiempo normal de uso. El método no determina cuáles son las variables que influyen en la tasa de fallos, tarea que quedará en manos del analista, pero al menos la distribución de Weibull facilitará la identificación de aquellos y su consideración, aparte de disponer de una herramienta de predicción de comportamientos. Esta metodología es útil para aquellas empresas que desarrollan programas de mantenimiento preventivo de sus instalaciones.

1.1.1.1. Características generales

Sabemos que la tasa de fallos se puede escribir, en función de la fiabilidad, de la siguiente forma:

$$\lambda(t) = - \frac{\frac{d[R(t)]}{dt}}{R(t)} \quad (51)$$

$$ó R(t) = \exp \left[- \int_0^t \lambda(t) dt \right]$$

siendo:

$\lambda(t)$ - Tasa de fallos

$R(t)$ - Fiabilidad

$F(t)$ - Infiabilidad o Función acumulativa de fallos

t - Tiempo

En 1951 Weibull propuso que la expresión empírica más simple que podía representar una gran variedad de datos reales podía obtenerse escribiendo:

$$\int \lambda(t)dt = \left(\frac{t-t_0}{\eta}\right)^\beta \quad (52)$$

por lo que la fiabilidad será:

$$R(t) = e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (53)$$

siendo :

γ - parámetro inicial de localización

η - parámetro de escala o vida característica

β - parámetro de forma

Se ha podido demostrar que gran cantidad de representaciones de fiabilidades reales pueden ser obtenidas a través de ésta ecuación, que como se mostrará, es de muy fácil aplicación.

La distribución de Weibull se representa normalmente por la función acumulativa de distribución de fallos $F(t)$:

$$F(t) = 1 - e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (54)$$

siendo la función densidad de probabilidad:

$$f(t) = \frac{\beta}{\eta^\beta} (t - \gamma)^{\beta-1} e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (55)$$

La tasa de fallos para esta distribución es:

$$z(t) = \frac{\beta}{\eta^\beta} (t - \gamma)^{\beta-1} \quad (56)$$

Las ecuaciones anteriores sólo se aplican para valores de $(t - \gamma) \geq 0$. Para valores de $(t - \gamma) < 0$, las funciones de densidad y la tasa de fallos valen 0. Las constantes que aparecen en las expresiones anteriores tienen una interpretación física.

- γ es el parámetro de posición (unidad de tiempos) o vida mínima y define el punto de partida u origen de la distribución.
- η es el parámetro de escala, extensión de la distribución a lo largo, del eje de los tiempos. Cuando $(t - \gamma) = \eta$ la fiabilidad viene dada por:
 $R(t) = \exp(-(1)^\beta) = 1/\exp(1^\beta) = 1 / 2,718 = 0,368$ (36,8%)
 Entonces la constante representa también el tiempo, medido a partir de $t_0 = 0$, según lo cual dado que $F(t) = 1 - 0,368 = 0,632$, el 63,2 % de la población se espera que falle, cualquiera que sea el valor de β ya que como hemos visto su valor no influye en los cálculos realizados. Por esta razón también se le llama usualmente vida característica.
- β es el parámetro de forma y representa la pendiente de la recta describiendo el grado de variación de la tasa de fallos.

Las variaciones de la densidad de probabilidad, tasa de fallos y función acumulativa de fallos en función del tiempo para los distintos valores de β , están representados gráficamente en la figura siguiente.

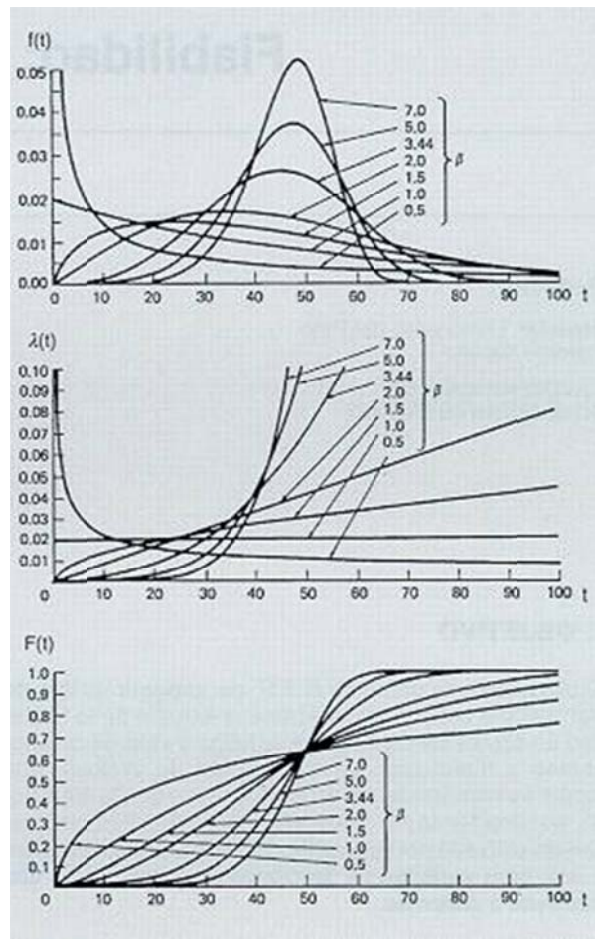


Figura 21: Variación de la densidad de probabilidad $f(t)$, tasa de fallos $\lambda(t)$ y la función acumulativa de fallos $F(t)$ en función del tiempo para distintos valores del parámetro de forma β

1.1.2. Representación de los modos de fallo mediante la distribución de weibull

En el estudio de la distribución se pueden dar las siguientes combinaciones de los parámetros de Weibull con mecanismos de fallo particulares:

$\gamma = 0$: el mecanismo no tiene una duración de fiabilidad intrínseca, y:

- si $\beta < 1$ la tasa de fallos disminuye con la edad sin llegar a cero, por lo que podemos suponer que nos encontramos en la juventud del componente con un margen de seguridad bajo, dando lugar a fallos por tensión de rotura.
- si $\beta = 1$ la tasa de fallo se mantiene constante siempre lo que nos indica una característica de fallos aleatoria o

pseudoaleatoria. En este caso nos encontramos que la distribución de Weibull es igual a la exponencial.

- si $\beta > 1$ la tasa de fallo se incrementa con la edad de forma continua lo que indica que los desgastes empiezan en el momento en que el mecanismo se pone en servicio.
- si $\beta = 3,44$ se cumple que la media es igual a la mediana y la distribución de Weibull es sensiblemente igual a la normal.

$\gamma > 0$: El mecanismo es intrínsecamente fiable desde el momento en que fue puesto en servicio hasta que $t = \gamma$, y además:

- si $\beta < 1$ hay fatiga u otro tipo de desgaste en el que la tasa de fallo disminuye con el tiempo después de un súbito incremento hasta t_0 ; valores de β bajos ($\sim 0,5$) pueden asociarse con ciclos de fatigas bajos y los valores de b más elevados ($\sim 0,8$) con ciclos más altos.
- si $\beta > 1$ hay una erosión o desgaste similar en la que la constante de duración de carga disminuye continuamente con el incremento de la carga.

$\gamma < 0$. Indica que el mecanismo fue utilizado o tuvo fallos antes de iniciar la toma de datos, de otro modo

- si $\beta < 1$ podría tratarse de un fallo de juventud antes de su puesta en servicio, como resultado de un margen de seguridad bajo.
- si $\beta > 1$ se trata de un desgaste por una disminución constante de la resistencia iniciado antes de su puesta en servicio, por ejemplo debido a una vida propia limitada que ha finalizado o era inadecuada.

1.1.3. Análisis de Weibull

Uno de los problemas fundamentales de la distribución de Weibull es la evaluación de los parámetros (γ, η, β) de esta distribución. Para ello se dispone de dos métodos: a través únicamente del cálculo mediante el método de los momentos o

el de máxima verosimilitud, en el que intervienen ecuaciones diferenciales difíciles de resolver, por lo que se utilizan poco, y mediante la resolución gráfica, que utiliza un papel a escala funcional llamado papel de Weibull o gráfico de Allen Plait que es el que vamos a desarrollar.

Resolución gráfica.

El papel de Weibull mostrado en la siguiente figura está graduado a escala funcional de la siguiente forma:

En el eje de ordenadas se tiene: $\ln \ln \frac{1}{1-F(t)}$ (Doble logaritmo neperiano).

En el eje de abscisas, tenemos: $\ln(t - \gamma)$.

Existen tres casos posibles en función del valor de γ .

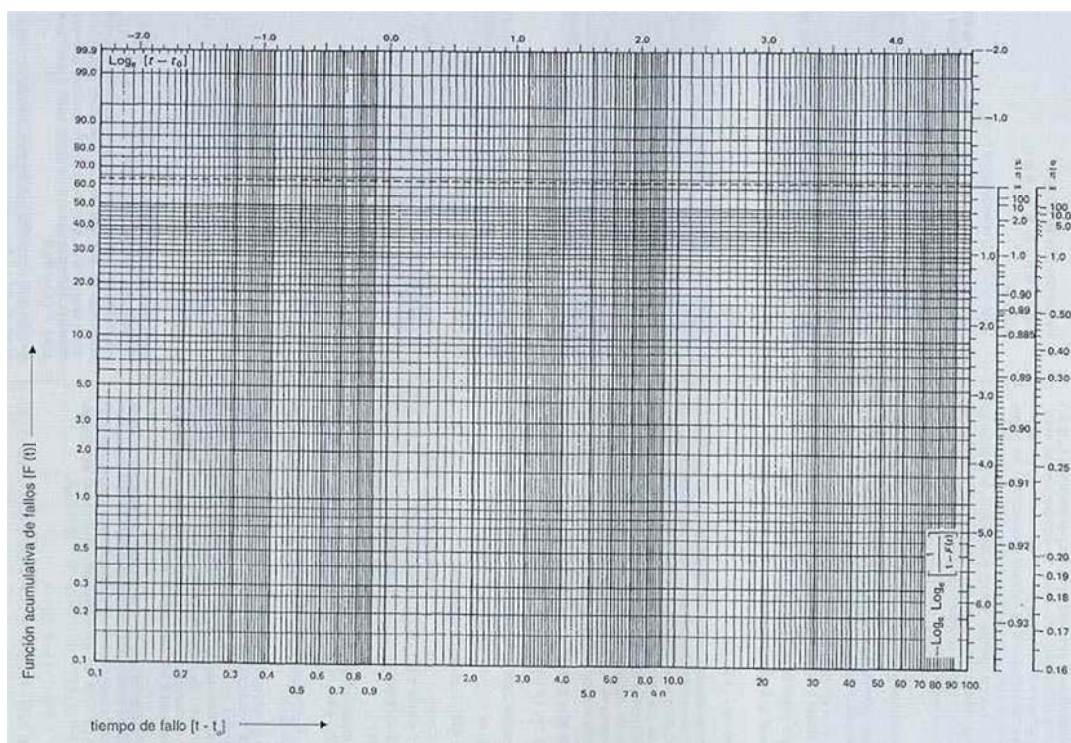


Figura 22: Muestra del papel de Weibull

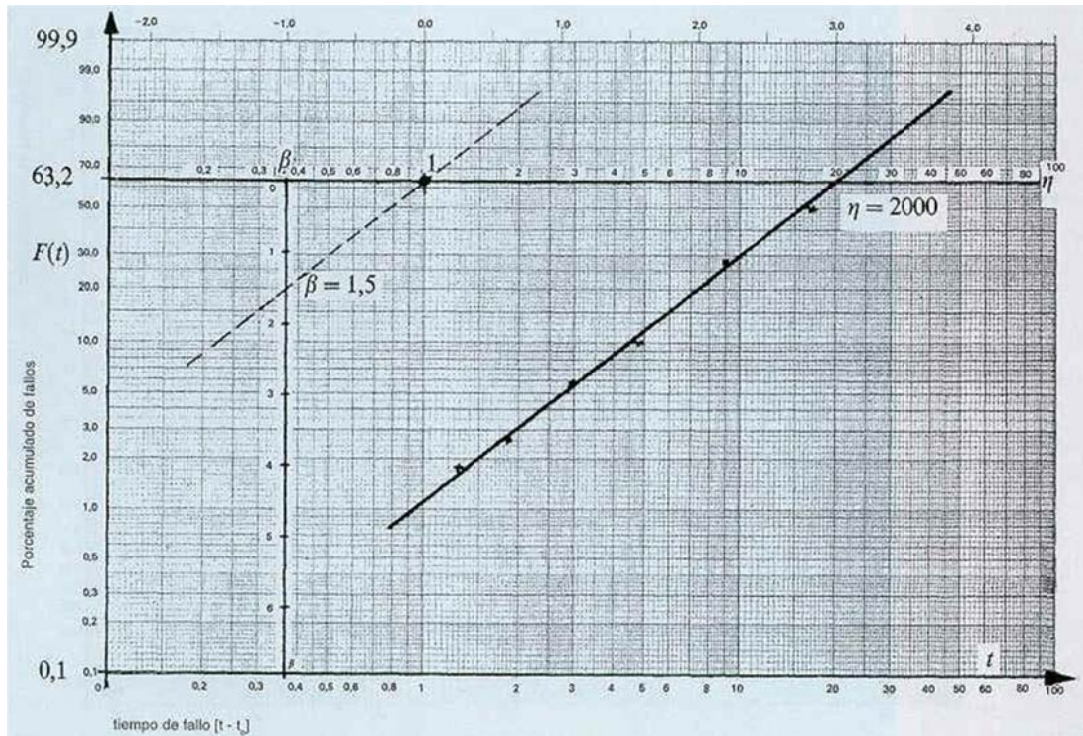


Figura 23: Lectura de los parámetros η y β en el papel de Weibull

Caso de $\gamma = 0$

Demostremos que cualquier grupo de datos que sigan la distribución de Weibull se pueden representar por una línea recta en el papel de Weibull. Partimos de la hipótesis de que el origen es perfectamente conocido y que coincide con los datos experimentales. Desde el punto de vista matemático partimos de la fórmula que nos relaciona la fiabilidad con la infiabilidad y teniendo en cuenta la expresión:

$$R(t) = \frac{1}{1 - F(t)} = e^{\left(\frac{1}{\eta}\right)^\beta} \quad (57)$$

Tomando logaritmos neperianos por dos veces:

$$\ln \ln \frac{1}{1 - F(t)} = \beta \ln t - \beta \ln \eta \quad (58)$$

Si a esta igualdad le aplicamos

$X = \ln t$ (variable función de t).

$$Y = \ln \ln \frac{1}{1-F(t)} \text{ (función de t).}$$

$$B = -\beta \ln \eta \text{ (constante).}$$

$$A = \beta \text{ (coeficiente director).}$$

de donde tenemos:

$$Y = AX + B \text{ (ecuación de una recta)}$$

Para determinar los parámetros β y η se utiliza la representación de Weibull.

- Cálculo de β : β es el parámetro de forma y representa la pendiente de la recta. Para calcularlo, se hace pasar una recta paralela a la recta obtenida con la representación gráfica de los datos de partida por el punto 1 de abscisas y 63,2 de ordenadas pudiendo leer directamente el valor de β en una escala tabulada de 0 a 7.
- Cálculo de η : η es el parámetro de escala y su valor viene dado por la intersección de la recta trazada con la línea paralela al eje de abscisas correspondiente al 63,2% de fallos acumulados. En efecto se demuestra que para la ordenada $\gamma = 0$, $F(t) = 63,2$.

- $Y = \ln \ln \frac{1}{1-F(t)} = 0$

- $\ln \frac{1}{1-F(t)} = 1; \ln \frac{1}{1-F(t)} = e; \ln \frac{1}{1-F(t)} = 1/e$

- $F(t) = 1 - [1/e] = 1 - [1/2,7183] = 1 - 0,3679 = 0,6321 \text{ (63,21 \%)}$

- De donde para $\gamma = 0$ tendremos que $AX + B = 0$; como según hemos visto anteriormente:

- $A = \beta \quad B = -\beta \ln \eta$

- Tendremos que se cumple:

- $\beta X - \beta \ln \eta = 0; \beta X = \beta \ln \eta;$

- $X = \text{Ln } \eta$
- Como $X = \text{Ln } t$, tenemos que $t = \eta$.
- η es el valor leído directamente en el gráfico de Allen Plait para la ordenada 63,2%, ya que la escala de abcisas está como ya se ha indicado en $\text{Ln } t$.
- Tiempo medio entre fallos (MTBF) o media: el tiempo medio entre fallos o vida media se calcula con la ayuda de la tabla 1, que nos da los valores de gamma y vale:
- $E(t) = \text{MTBF} = \eta \Gamma(1 + 1/\beta)$
- Desviación estándar o variancia s: se calcula también con la ayuda de la tabla 1 y vale:
- $(s/\eta)^2 = \Gamma(1 + 2/\beta) - [\Gamma(1 + 1/\beta)]^2$

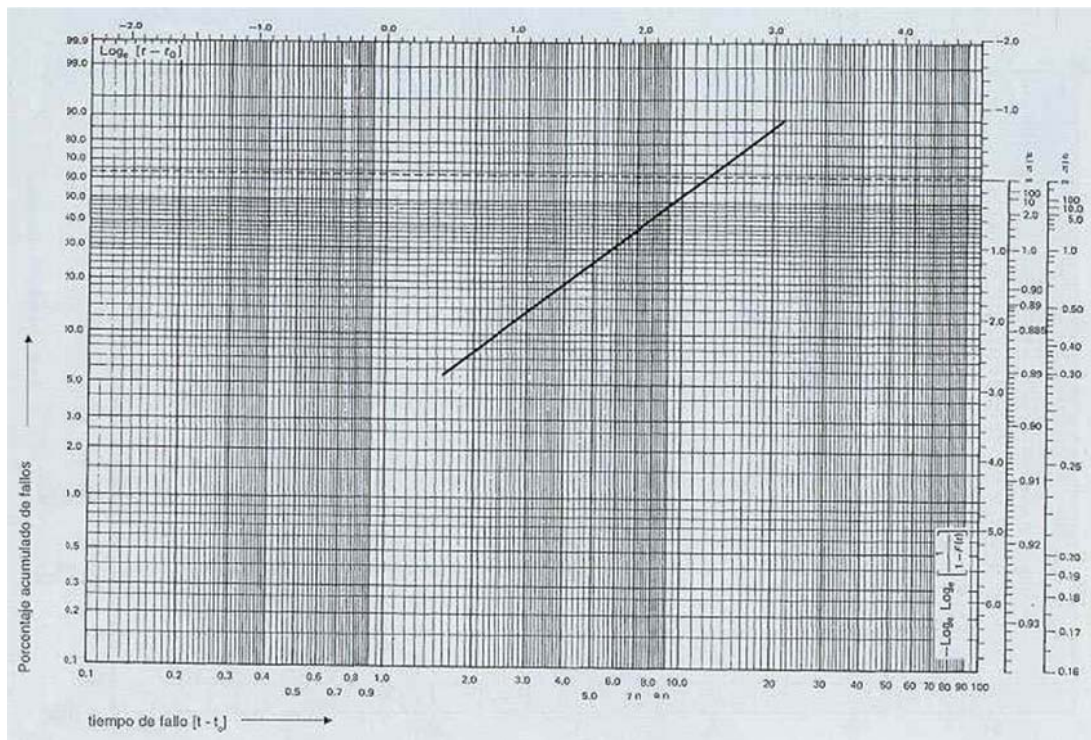


Figura 24: Ejemplo representación Weibull

3.4.8. Distribución Gamma

Función de densidad

$$f(t) = \frac{\lambda}{\Gamma(\tau)} (\lambda t)^{\tau-1} e^{-\lambda t} \quad (59)$$

Función de distribución

$$F(t) = \frac{\lambda}{\Gamma(\tau)} \int_0^t (\lambda x)^{\tau-1} e^{-\lambda x} dx \quad (60)$$

Fiabilidad

$$R(t) = \frac{\lambda}{\Gamma(\tau)} \int_t^{\infty} (\lambda x)^{\tau-1} e^{-\lambda x} dx \quad (61)$$

Vida media

$$\theta = E(\Gamma) = \frac{\tau}{\lambda} \quad (62)$$

Tasa de fallo

$$z(t) = \frac{t^{\tau-1} e^{-\lambda t}}{\int_t^{\infty} x^{\tau-1} e^{-\lambda x} dx} \quad (63)$$

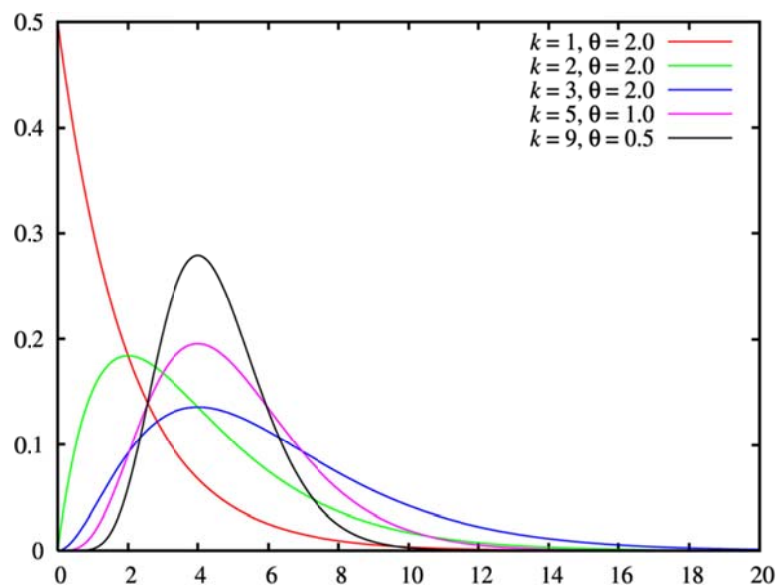


Figura 25: Distribución Gamma

3.4.9. Fallos por degradación de los componentes

Un componente sufre un fallo por degradación cuando alguno de sus parámetros de actuación deriva gradualmente hasta salirse de unas tolerancias fuera de las cuales el funcionamiento se considera incorrecto.

Un parámetro al que se llamará $\zeta(t)$, para una población dada de componentes nominalmente idénticos, se distribuirá con una densidad de probabilidad función del tiempo t . Las curvas de densidad exhibirán una deriva del valor medio $E(\zeta, t)$ y un aumento de la desviación típica $\sigma(t)$. Los límites 3σ , es decir, $E(\zeta, t) + 3\sigma(t)$ y $E(\zeta, t) - 3\sigma(t)$ se irán separando, haciéndose la curva de densidad cada vez más aplanada. Esto se ha representado en la figura deriva de un parámetro, donde tres curvas de densidad correspondientes a los instantes 0, t_1 , y t_2 se han dibujado abatidas sobre el plano t - $z(t)$, siendo $z(t)$ los valores que puede tomar el parámetro $\zeta(t)$. Si z_l y z_s son, respectivamente, los límites inferior y superior de tolerancia del parámetro, se ve que en el instante inicial los límites 3σ quedan dentro de dichos límites, indicando que prácticamente toda la población de componentes es buena. En el instante t_1 , el límite 3σ superior alcanza el valor z_s , mientras que el límite 3σ inferior ha descendido perceptiblemente de z_l . En el instante t_2 , ambos límites se han salido de tolerancias. Las áreas rayadas indican la fracción de los componentes que aun se mantiene dentro de tolerancias.

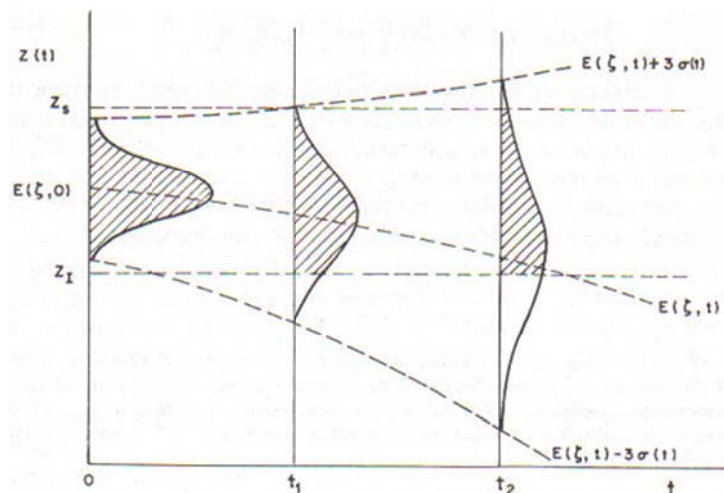


Figura 26: Deriva de un parámetro

La probabilidad de que en un instante cualquiera t un componente elegido al azar esté dentro de las tolerancias será:

$$R_D(t) = \int_{z_l}^{z_s} f(z, t) dz \quad (64)$$

3.5. Ensayos de fiabilidad

3.5.1. Introducción

A lo largo de las etapas de definición, diseño, fabricación de prototipos, producción y uso de un componente, equipo o sistema, es necesario realizar siempre gran cantidad de ensayos conocidos tradicionalmente como ensayos de factibilidad del diseño, desarrollo, calificación, control de calidad, aceptación y evaluación en servicio. Todos ellos son básicamente ensayos de actuaciones, simples pruebas funcionales o ensayos ambientales. Sirven para determinar si un dispositivo funciona correctamente y es capaz de soportar determinados esfuerzos internos y ambientales (temperatura, humedad, vibración, choque, etc.). Evidentemente, los fallos que se experimenten en cualquier ensayo pueden revelar, y revelan a menudo, puntos de fiabilidad insatisfactoria que deben dar lugar a una acción correctiva (mejora del diseño, de los métodos de producción, etc.), pero no bastan, por lo general, para establecer conclusiones cuantitativas sobre la fiabilidad del dispositivo.

En los últimos años, a los ensayos tradicionales se han añadido los ensayos de fiabilidad, encaminados específicamente a la obtención de datos cualitativos y cuantitativos de fiabilidad.

3.5.2. Clases de ensayos de fiabilidad

Los ensayos de fiabilidad pueden clasificarse en: ensayos de medición y demostración de la fiabilidad, ensayos de medición de la variación de parámetros, ensayos de investigación de modos y mecanismos de fallo y ensayos de cribado de componentes.

1.1.1. Ensayos de medición y demostración

Por la medición de la fiabilidad se entiende la estimación estadística de la misma o de los parámetros que la determinan. Cuando existe un requisito previo de fiabilidad, una contrastación estadística puede demostrar, con determinado nivel de confianza, que el dispositivo lo cumple.

Los dispositivos a ensayar para medir o demostrar su fiabilidad pueden ser del tipo que sólo se usa una vez (dispositivos *one-shot*). Dentro de este tipo hay dispositivos cuya función implica su destrucción, de forma que no pueden someterse a un ensayo previo a su utilización, como ocurre con los fusibles, elementos pirotécnicos

(iniciadores, corta-cables, etc.) y cohetes de combustible sólido. En este caso, la medición o demostración de la fiabilidad exige un muestreo por atributos (éxito-fracaso) o por variables (es decir, por medidas). Otros dispositivos *one-shot* pueden ensayarse antes de su empleo, como los sistemas de despliegue de los mástiles de satélite y los sistemas de guiado de misiles. Las piezas cuya fiabilidad puede estudiarse mediante un modelo esfuerzo-resistencia pueden ensayarse a rotura para determinar la distribución de su resistencia, utilizando un plan de muestreo por variables. Otras veces, se ensayan hasta un nivel de esfuerzo de prueba mediante un plan por atributos. Los planes por atributos exigen una muestra mayor que los planes por variables, pero en el caso de las piezas mecánicas, las supervivientes pueden usarse posteriormente.

Los dispositivos cuya fiabilidad depende del tiempo se someten a ensayos de vida. Cuando se conoce la forma de la distribución de tiempos hasta el fallo o tiempos entre fallos, es posible establecer planes de muestreo por variables para estimar o contrastar los parámetros desconocidos. Si no es posible establecer una hipótesis de distribución, puede estimarse o contrastarse la fiabilidad para uno o más tiempos de funcionamiento dados (método no paramétrico por atributos).

Los métodos de muestreo por atributos, son muy familiares en control de calidad. El muestreo por variables usado en control de calidad suele basarse en la hipótesis de que el parámetro que interesa se distribuye normalmente.

Aunque los datos estadísticos recogidos en campo de utilización pueden servir en muchos casos para medir la fiabilidad *a posteriori*, llegan demasiado tarde, para ciertos fines importantes, como es la mejora de la fiabilidad por eliminación de puntos débiles desde las primeras unidades de producción. Por otra parte, muchos contratos obligan hoy día a una demostración de la fiabilidad previa a la entrega del producto. Así como la medición de la fiabilidad puede realizarse *post mortem*, mediante datos de utilización real, a fin de obtener una información válida para futuras aplicaciones, la demostración suele implicar la realización de ensayos de laboratorio diseñados como un experimento estadístico cuidadoso. Cuando se pretenda utilizar datos de campo de utilización a fines de inferencia estadística, es necesario analizar dichos datos para ver si pueden considerarse resultado de un muestreo aleatorio, es decir, si son una buena representación de la población de dispositivos o, por el contrario, son indeseables. Los métodos estadísticos a emplear deben ser definidos por un especialista.

La necesidad de realizar ensayos de fiabilidad con aquellos tipos de dispositivos cuya fiabilidad no puede predecirse mediante un modelo matemático, con datos previos de la fiabilidad de sus subconjuntos o piezas, es evidente. Esto ocurre con los componentes más simples. Pero cuando se trata de niveles de montaje más elevados, sólo los ensayos podrán darnos en un plazo relativamente corto una indicación de lo acertado de la predicción, de si los datos de fiabilidad de componentes usados en dicha predicción eran correctos para la aplicación de los mismos, si aparecen modos de fallo no previstos y si el modelo matemático era adecuado.

A veces se requiere un análisis de *crecimiento de la fiabilidad* como función permanente a través de las diferentes etapas de evolución del producto. Normalmente, la fiabilidad mejora durante las fases de desarrollo y ensayos, sufriendo un descenso en la fase inicial de empleo para alcanzar sus mayores valores cuando el dispositivo ha alcanzado su madurez en campo de utilización. Para estudiar el crecimiento de la fiabilidad se requiere hacer sucesivas mediciones de la misma, preferiblemente usando técnicas de estimación por intervalos de confianza.

1.1.2. Ensayos de variación de parámetros

Para la predicción de la fiabilidad respecto de fallos por deriva de parámetros es necesario conocer cómo varían los parámetros de los componentes por efecto del envejecimiento, que es función de los esfuerzos y del tiempo. Los ensayos de variación de parámetros de los componentes consisten en mediciones sucesivas de los parámetros a lo largo de extensos períodos de funcionamiento. La variación que se observe entre los valores de los parámetros de las unidades de una muestra se traducirán en distribuciones estadísticas.

También debe mencionarse el análisis, mediante ensayos, de la sensibilidad de los parámetros de salida a la variación de los parámetros internos.

Los fallos por deriva de parámetros se suelen incluir en los ensayos de medición y demostración definiendo con precisión tolerancias para los parámetros de salida, lo que equivale a definir los fallos no catastróficos.

1.1.3. Ensayos de investigación de modos y mecanismos de fallo

Con estos ensayos se pretende conocer mejor las causas físicas de fallo de los componentes para mejorar su diseño. Es frecuente aprovechar para estos análisis los fallos ocurridos en ensayos realizados con otros fines, así como los fallos en campo de

utilización. Para ello es preciso establecer un buen sistema de informes de fallos. Cuando, a fin de conseguir suficientes fallos en poco tiempo, se aplican factores de aceleración, es preciso tener en cuenta que la elevación del nivel de esfuerzos internos y ambientales puede modificar los modos y mecanismos de fallo, haciendo aparecer algunos que no se observan en utilización normal e incluso eliminando parcial o totalmente otros que se producen a niveles normales de esfuerzos.

1.1.4. Ensayos de cribado

El cribado o selección unitaria (*screening*) se practica de acuerdo con las especificaciones para componentes de alta fiabilidad. La criba tiene por objeto eliminar las unidades con taras infantiles y aquellas que presentan síntomas indicativos de vida corta. Se efectúa sometiendo a *toda la población* a una serie de pruebas e inspecciones. Con ello se espera (y la experiencia indica que se consigue) retener a los individuos más aptos para el servicio. La extensión del cribado puede reducirse a un solo ensayo o abarcar una serie considerable de pruebas e inspecciones, cada uno de cuyos pasos es eliminatorio.

El ensayo más típico dentro de los incluidos en las cribas es la purga (*burn-in*) que consiste en someter al lote a un funcionamiento de envejecimiento para inducir fallos infantiles. Naturalmente, los componentes que no acusen inicialmente una tasa de fallo decreciente no deben someterse a purga. Los esfuerzos internos y ambientales suelen ser los máximos esperados en funcionamiento normal o los nominales. La duración de la purga para componentes electrónicos suele ser de una semana (168 horas) o algo más. En determinadas circunstancias se han aconsejado purgas de mayor duración. Evidentemente, si la misión del dispositivo debe durar h horas, el tiempo óptimo de purga t_0 debe ser tal que la fiabilidad de misión sea máxima:

$$Máx(R) = Máx \left(e^{-\int_{t_0}^{t_0+h} z(t) dt} \right) \quad (65)$$

lo que equivale a hallar el valor de t_0 que hace mínimo el valor absoluto del exponente:

$$Mín \left(\int_{t_0}^{t_0+h} z(t) dt \right) \quad (66)$$

Para poder determinar t_0 hace falta conocer la función $z(t)$ de este tipo de componente con una precisión difícil de conseguir.

Se ha llamado purga al *burn-in* por su aspecto de eliminación de "individuos subnormales". Este aspecto es común a cualquier inspección o prueba dentro de una criba. Pero la purga suele cumplir también una función de estabilización de características. Los franceses llaman al *burn-in* rodaje, pero el rodaje (en inglés *run-in* o *break-in*) tiene una misión ajena a la criba. No elimina individuos, sino defectos y asperezas, con vistas a un funcionamiento posterior "suave" y mejor.

Aparte de eliminar las unidades falladas en las pruebas, la criba desecha aquellas que en las inspecciones y mediciones de parámetros revelan condiciones inadecuadas (fuera de tolerancias) o indicaciones de bajo potencial de tiempo de funcionamiento correcto.

Al desechar las unidades con deriva excesiva y las pertenecientes a lotes con alta proporción de deriva excesiva, la criba nos deja un conjunto de unidades mucho más estable y homogéneo.

3.5.3. Ensayos de vida de larga duración

Anteriormente se ha dicho que los dispositivos cuya fiabilidad depende del tiempo se someten a ensayos de vida para medir o demostrar su fiabilidad. Este tipo de ensayos es largo y costoso, pues exige mantener en funcionamiento, en condiciones ambientales específicas, una muestra hasta que fallen todas o parte de las unidades. (En ciertos casos, puede demostrarse un requisito de fiabilidad sin fallos, pero el tiempo acumulado de ensayo no deja de ser muy largo si la fiabilidad a demostrar es elevada y el nivel de confianza también.)

Si se puede aceptar razonablemente una hipótesis sobre la forma de la distribución de tiempos hasta el fallo (o entre fallos) para el dispositivo que nos interesa, la finalidad de nuestro ensayo será estimar los parámetros de la distribución o contrastar estadísticamente la hipótesis de que dichos parámetros satisfacen determinadas condiciones. En caso de no poder admitir una hipótesis de distribución, los ensayos no paramétricos (estimación o contrastación de puntos de la curva de fiabilidad) resultan todavía más costosos.

Así como las pruebas funcionales de ingeniería se hacen en tiempo corto y en condiciones ambientales normales, los ensayos ambientales tradicionales exigen

generalmente condiciones máximas de temperatura, presión o vacío, choque, vibración, aceleración, etc. Los ensayos de fiabilidad, los más largos de todos, exigen condiciones ambientales similares a las esperadas en utilización real.

1.1.1. Hipótesis de distribución

De las numerosas distribuciones que pueden representar el tiempo hasta el fallo o el tiempo entre fallos de los dispositivos, las más usuales son la exponencial, la normal, la log-normal, la de Weibull y la gamma.

La validez de los ensayos basados en una hipótesis de distribución depende de la buena elección de dicha hipótesis. A veces, la experiencia con dispositivos análogos puede justificar esta elección. Cuando se dispone o se puede disponer de resultados de ensayo, suelen utilizarse procedimientos gráficos con papel probabilístico (como papel de Weibull). Los métodos analíticos basados en contrastes de bondad de ajuste son muy utilizados. En todo caso, tanto los métodos gráficos como analíticos no consiguen sino descartar las hipótesis que parecen más disparatadas. De esta forma, no es raro ver, por ejemplo, que a unos datos de ensayo se puede ajustar una distribución normal, una de Weibull y una gamma con el resultado de que ninguna de las tres hipótesis parezca inverosímil. Continuar ensayando puede ayudar a descartar alguna. En todo caso, es en las colas donde se revelan más las discrepancias y conseguir datos suficientes para poder opinar en puntos alejados de la media, es un proceso largo y costoso.

1.1.2. Esfuerzos y condiciones ambientales

Para conocer la fiabilidad de un dispositivo sometido a determinados esfuerzos y condiciones ambientales, es necesario ensayarlo con esos esfuerzos y en ese ambiente. La extrapolación de resultados obtenidos en condiciones diferentes es arriesgada. Por ello, parecerá ideal realizar el ensayo en campo de utilización real. No obstante, suele ser más práctico realizar los ensayos en laboratorio, aplicando los esfuerzos internos especificados y simulando el ambiente de aplicación real. Habrá que recurrir a los ensayos en ambiente operativo cuando este ambiente no pueda simularse en laboratorio.

Cuando se trata de investigar el efecto de los factores de alivio (*derating*) para componentes, es preciso efectuar ensayos a diferentes niveles de esfuerzo y ambientales. Por el contrario, la demostración de fiabilidad se limitará a los esfuerzos y ambientes más severos esperados en la vida operativa del dispositivo.

Las especificaciones de demostración de la fiabilidad deben contener detalles completos sobre los esfuerzos y niveles ambientales a aplicar durante los ensayos. En muchas ocasiones, la aplicación de esfuerzos internos y ambientales se realiza en forma de ciclos. Véase, la norma MIL-STD-781.

3.5.4. Ensayos acelerados

Se entiende por aceleración de un ensayo de vida al aumento de la exposición al fallo por unidad de tiempo. Este aumento se puede conseguir elevando el nivel de esfuerzos internos, la severidad ambiental o la frecuencia de aplicación de estas sollicitaciones.

Cuando se elevan los esfuerzos y la severidad de las condiciones ambientales, aceleramos la producción de fallos tanto por azar como por desgaste (envejecimiento). En estas condiciones, es necesario mantener los esfuerzos y parámetros ambientales dentro de ciertos límites para no cambiar los mecanismos de fallo. Se pretende que todo ocurra igual que en condiciones normales, aunque en menos tiempo. Esto es lo que se llama *aceleración verdadera* y es muy difícil o imposible de conseguir en la generalidad de los casos con dispositivos afectados de múltiples mecanismos de fallo. En ocasiones, es posible acelerar por separado un mecanismo de fallo (como la fatiga), pero en electrónica el problema suele ser más complejo.

Se ha hablado mucho de las ecuaciones de aceleración, algunas tomadas de la física, como la de Arrhenius o la de Eyring. Con estas ecuaciones u otras, se pretende correlacionar el comportamiento de un dispositivo con la temperatura, generalmente. Dada la gran variedad de componentes, aún sin que se salga del dominio de la electrónica, el campo de estudio es enorme y lo que hasta ahora se sabe es muy poco para establecer predicciones válidas en la generalidad de los casos.

En electrónica se han realizado numerosos ensayos para correlacionar la tasa de fallo con uno o más tipos de sollicitación. Por ejemplo, se ha tratado de determinar *factores de aceleración* para la tasa de fallo en función de la tensión eléctrica y de la temperatura.

Los ensayos con sollicitaciones escalonadas (*step-stress*) suelen ser muy útiles para la investigación de mecanismos de fallo, pero no suministran en general datos de tasa de fallo.

3.5.5. Introducción a la ecuación de Arrhenius

El modelo de Arrhenius es probablemente la relación más común utilizada en los ensayos acelerados de vida, donde la variable de aceleración es la temperatura. El modelo se describe a continuación:

$$R(T) = Ae^{-\frac{E_A}{KT}}$$

donde:

R, es la velocidad de reacción.

T, es el valor absoluto de la temperatura (kelvin).

A, es uno de los parámetros no térmicos del modelo que debe ser determinado.

E_A , es la energía de activación (eV).

K, es la constante de Boltzman = $8.617385 \cdot 10^{-5} \text{ eVK}^{-1}$.

La energía de activación es la energía que una molécula debe tener para participar en la reacción. En otras palabras, es una medida del efecto que la temperatura tiene sobre la reacción.

Para que el modelo de Arrhenius sea válido para ensayos en células solares que es el objetivo de este proyecto hay que tener en cuenta que el ensayo ha de realizarse a diferentes temperaturas, dado que hay dos incógnitas en la ecuación, la energía de activación del fallo y la constante A. La diferencia entre esas temperaturas ha de ser considerable, para poder evaluar la energía de activación de una forma precisa, teniendo en cuenta de no usar una temperatura que provoque modos de fallos. El tiempo en que se produce un fallo es inversamente proporcional a la velocidad de reacción:

$$t(T) = Ae^{+\frac{E_A}{KT}}; \quad (67)$$

Esta ecuación se puede linearizar tomando logaritmos neperianos a ambos lados:

$$\ln t(T) = \ln A + \frac{E_A}{KT} \quad (68)$$

Representado una variable temporal de la vida del dispositivo con respecto $1/KT$ a varias temperaturas tendremos puntos que siguen una línea recta de la que podremos obtener la energía de activación. Una vez obtenida la energía de activación es posible obtener la ley de vida a la temperatura nominal de funcionamiento.

Este es el modelo que utilizaremos en los ensayos acelerados de los que mostraremos unos resultados preliminares en el capítulo de resultados.

4 Sistema de medida

La realización de ensayos acelerados en células solares presenta importantes dificultades por la necesidad de hacer funcionar a las células dentro de la cámara climática. A la mayoría de los dispositivos es sencillo hacerlos trabajar dentro de la cámara climática, sin embargo la iluminación de las células dentro de la cámara climática presenta importantes dificultades. En este capítulo vamos a explicar la metodología empleada para realizar los ensayos acelerados en células solares de concentración.

4.1. Metodología de trabajo

A continuación se describen los pasos llevados a cabo para la realización de la metodología para el ensayo acelerado en células solares, para posteriormente poder realizar la evaluación de la fiabilidad de las células solares de concentración. Para realizar los ensayos acelerados es necesario hacer funcionar y caracterizar a la célula dentro de la cámara climática. El funcionamiento de la célula se emulará mediante la inyección de corriente mientras que su caracterización se realizará mediante la curva I/V en oscuridad. La emulación de funcionamiento y la caracterización de la célula serán explicadas en detalle en este mismo capítulo más adelante.

Otro tema que hay que evaluar es la temperatura a la que se puede realizar los ensayos acelerados. La aceleración de los ensayos aumenta con la temperatura y por lo tanto es necesario determinar la temperatura máxima del encapsulado, así podremos conocer cuál es la temperatura máxima a la que podemos someter a las células durante el ensayo, teniendo en cuenta la temperatura tanto de la cámara climática como de la propia célula al ser sometida a las condiciones de funcionamiento inyectándola corriente.

El siguiente paso será diseñar el hardware y el software necesarios para poder llevar a cabo el ensayo con las condiciones de funcionamiento descritas. El estudio de este Trabajo Fin de Máster se basa principalmente en este punto.

Posteriormente hay que definir el número de células a estudio, dicho número ha de ser un número razonable para poder determinar la ley de vida y el motivo del fallo. En nuestro caso se ha desarrollado un software preparado para estudiar hasta 40 células dado que es el límite alcanzado con el hardware facilitado. A la finalización del periodo de diseño y puesta en funcionamiento del sistema se ha dejado funcionando con 20 células conectadas para la realización del primer ensayo.

Además de todo esto es necesario determinar las técnicas de caracterización que van a ser empleadas para la realización del test, definir cuál es el criterio de fallo, debemos determinar qué consideraremos fallo para posteriormente evaluar que células han fallado.

Una vez realizado todo el montaje, se procederá a realizar los ensayos, para ello es necesario caracterizar previamente todas y cada una de las células solares utilizadas para el ensayo, para comprobar la evolución de cada una de ellas.

Se pretende dividir las células en 3 cámaras distintas, para someter a cada uno de los grupos a una temperatura distinta, pudiendo determinar así el efecto de aceleración que causa la temperatura sobre la ley de vida de la célula solar.

Una vez terminado el ensayo, caracterizar la célula solar con los datos obtenidos. Se caracterizará la potencia generada por cada célula durante el ensayo acelerado en temperatura. Para poder caracterizar la célula dentro de la cámara climática se ha empleado un modelo con el que podremos evaluar la curva I-V en iluminación a partir de la curva I-V en oscuridad obtenida periódicamente dentro de la cámara.

Y para finalizar hay que realizar el análisis de fiabilidad. Para el análisis de la fiabilidad asumiremos la distribución de Weibull como función de distribución de fallos. Los ensayos se realizarán a varias temperaturas por lo que emplearemos el modelo de Arrhenius para la evaluación del factor de aceleración.

A continuación vamos a explicar la emulación de las condiciones de funcionamiento y la caracterización de las prestaciones de la célula dentro de la cámara climática.

4.2. Condiciones de funcionamiento

Para la realización de ensayos acelerados sobre semiconductores en la gran mayoría de las situaciones el factor de estrés utilizado es la temperatura, dado que la degradación del dispositivo se acelera con la temperatura siguiendo la ley de Arrhenius. Esta ley está basada en los procesos físico/químicos que ocurren en los semiconductores y relaciona el tiempo transcurrido de ensayo a alta temperatura, con el tiempo que habría transcurrido a temperatura nominal de trabajo.

El problema de simular las condiciones de trabajo nominales, es aplicar luz solar simulada y concentrada a las células dentro de la cámara climática a alta temperatura. Se podrían plantear algunas soluciones, entre ellas se plantearon las siguientes:

- Concentrar la luz sobre cada célula introduciendo una guía de luz dentro de la cámara climática para cada célula. La fuente de luz puede ser un láser por cada unión de la célula de concentración III-V, ya que este tipo de dispositivo puede alcanzar la irradiancia equivalente a una concentración de 1000 soles. Este procedimiento sólo es válido para células de pequeña área, del orden de 1 mm², e implica un costo en equipamiento muy elevado.
- Otra alternativa valorada es introducir en la cámara climática los módulos solares completos con sus lentes de concentración, e iluminar a una concentración de un sol desde el exterior a través de una pared transparente de la cámara, mediante lámparas continuas de xenón. Esto implica cámaras grandes, la calibración periódica de la luz suministrada por las lámparas, que se degradan y deterioran a lo largo de las miles de horas de ensayo. Por otro lado, realizar los ensayos acelerados en temperatura en módulos solares completos limita de forma importante la temperatura del ensayo, y por lo tanto el factor de aceleración, ya que muchos de los materiales (adhesivos, plásticos) se degradan de forma importante por encima de los 100-120°C.

Tras el previo descarte de las dos opciones mencionados, en este proyecto se evalúa e implementa una nueva metodología para realizar los ensayos acelerados en temperatura en células solares, puesto que como hemos visto iluminar la célula solar dentro de la cámara climática presenta dificultades muy importantes se ha optado por simular el funcionamiento de la célula dentro de la cámara. Para ello se han simulado las condiciones eléctricas de trabajo mediante la inyección de una corriente en directa a la célula. La determinación del valor de dicha corriente directa la ha implementado el equipo del IES mediante la caracterización de las células solares con las que una vez implementado este trabajo fin de máster realizarán los ensayos.

En el caso particular de las células bajo ensayo, se les inyectará en directa con una corriente de 3,2A, Esta corriente se ha calculado por tener la misma densidad de corriente que tienen las células en condiciones nominales de irradiancia. Cuando haya

pasado el tiempo establecido de estrés, las células se dejan en reposo el tiempo necesario para que se estabilicen en temperatura. Posteriormente se realizarán las medidas de tensión y corriente que nos permitan obtener la curva I-V en oscuridad.

El proceso acelerado de degradación producido en la estructura semiconductor por la corriente directa inyectada debe ser equivalente a la producida por la radiación solar en condiciones reales de trabajo debido a la corriente generada en la zona activa. Hay que tener en cuenta dos consideraciones, que nuestro ensayo no simula la degradación producida por la radiación solar que incide sobre la célula, sin embargo es previsible que esta degradación no sea significativa a nivel de semiconductor ya que las células III-V son más robustas que las de silicio frente a la radiación. Y la segunda consideración que si medimos en iluminación la célula fuera de la cámara una vez realizado el ensayo, no podremos observar toda la degradación sufrida por la célula, ya que el camino de la corriente en iluminación no es el mismo que la zona más degradada con la inyección de corriente directa.

Por lo tanto, para poder caracterizar la célula solar dentro de la cámara climática se usará un modelo con el que se puede evaluar la curva I-V en iluminación, a partir de la curva I-V en oscuridad que se mide periódicamente dentro de la cámara climática. Por lo tanto, dentro de la metodología de este proyecto será necesario tomar la medida de la curva IV en oscuridad cada X tiempo, como se explicará en el desarrollo software, dicho valor dependerá de la aceleración del ensayo y por lo tanto será configurable en función de las necesidades del ensayo.

4.3. Potencia generada por la célula

La caracterización de la célula se obtiene mediante la curva I-V en oscuridad a la temperatura del ensayo de vida acelerado. Con esta información vamos a obtener información sobre el modo de degradación de la célula y la evolución de la máxima potencia.

Para obtener información de las características de la célula dentro de la cámara climática se parte de la curva I-V en oscuridad para a partir de ella poder evaluar la curva I-V en iluminación, una vez obtenida esta se obtiene el punto de máxima potencia generada P_{max} . Para esto se tomará como referencia para cada célula la medida del valor de potencia generada a la temperatura del ensayo antes de comenzar el mismo, P_0 ó P_{max} en $t=0$.

Para obtener la curva I-v en iluminación a partir de la curva I-V en oscuridad hay que emplear el principio de superposición, que fue probado en la década de los 70 para células de silicio monounión. Este método se ha comprobado que funciona muy bien cuando las células son de buena calidad.

El principio de superposición consiste en sumar las corrientes de las dos soluciones. Para un determinado nivel de tensión la corriente en iluminación es la suma del valor de la corriente en cortocircuito, I_{sc} , más el valor de la corriente en oscuridad, I_{dark} , para cada tensión. Ambas tensiones son de signo contrario, por lo que se restarán.

Para la corriente en oscuridad, se supone que no existe generación de corriente por iluminación, pero sí que existe una tensión en oscuridad, V_{dark} , en los terminales de la célula. Para la corriente generada en iluminación, asumimos que la corriente obtenida lo hace a una tensión de célula igual a cero, sin carga. De ahí el nombre I_{sc} , corriente en corto circuito (*short circuit*).

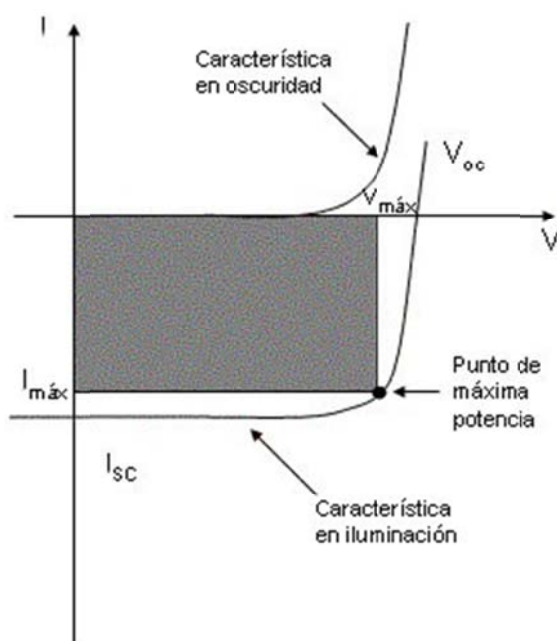


Figura 27: Curva en oscuridad y en iluminación para una célula solar.

Las ecuaciones de las curvas de iluminación (V_{ilum} , I_{ilum}) se obtiene a partir de las curvas en oscuridad (V_{dark} , I_{dark}) para un nivel de corriente fotogenerada I_L . Aplicando el principio de superposición y corrigiendo la tensión en la resistencia serie siguiendo las ecuaciones que se muestran a continuación:

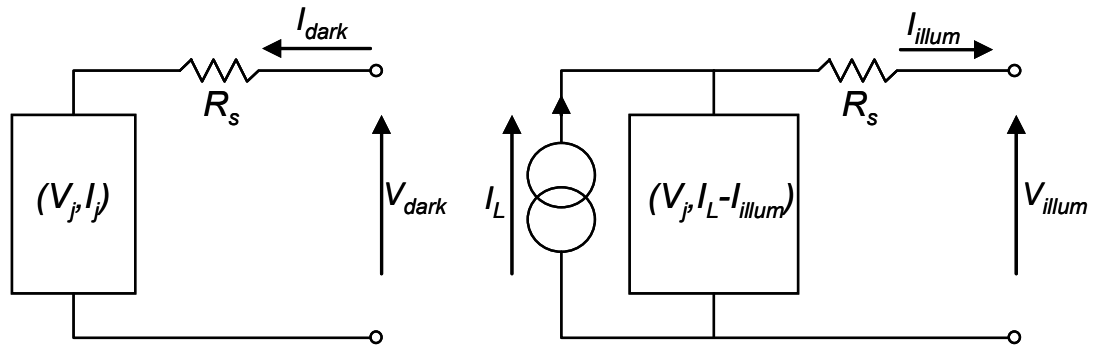


Figura 28: Modelo para calcular la curva I-V en iluminación a partir de la curva en oscuridad.

$$V_{illum} = V_{dark} - I_L R_s \quad (69)$$

$$I_{illum} = I_L - I_{dark} \quad (70)$$

Donde R_s es la resistencia serie.

Estas ecuaciones se emplean para obtener la curva I-V en iluminación, y a partir de ésta la potencia máxima generada por la célula a partir de la curva I-V en oscuridad a la temperatura del ensayo.

Una vez descrita la metodología general de los ensayos en los capítulos siguientes se va a explicar cómo se ha implementado tanto el software como el hardware para realizar los ensayos de forma automática.

5 Diagrama general y fases

5.1. Metodología de funcionamiento

Para poder llevar a cabo los ensayos descritos en los apartados anteriores es necesario simular las condiciones de funcionamiento de la célula, estas condiciones de funcionamiento se simularán sometiendo a estrés a la célula, dicho estrés se consigue inyectando una corriente de 3,2 Amperios, valor obtenido de la caracterización de las células, como se ha explicado en el apartado anterior. Para realizar los ensayos es necesario considerar:

- Después de ese tiempo de estrés es necesario un tiempo de reposo para asegurar la estabilización en temperatura de la célula, dado que al estar sometida a inyección de corriente la célula se calienta, con 10 minutos de estrés la célula alcanza los 70°C.
- Para comprobar la temperatura del horno a la hora de realizar las medidas, se introducirá junto con las células un sensor de temperatura.
- Para poder medir la célula es necesario que ésta no esté siendo sometida a ningún estrés, además de ser aislada del resto del circuito para poder tomar las medidas correspondientes.
- Las medidas como se explicará más adelante serán tomadas a 4 hilos, caracterizando así la curva IV, para obtener dicha curva, lo primero será aislar las células y posteriormente se realizará un barrido en tensión, tomando medidas de tensión y de corriente para cada tensión establecida.
- Para la medida de la curva I-V se usarán dos dispositivos distintos mencionados en el apartado hardware, con el primero de ellos aislaremos la célula en cuestión del resto del circuito y tomará la medida en tensión, y el segundo de ellos, establecerá el I tensión seleccionada por el usuario y medirá la corriente que hay en la célula.

En el apartado siguiente mostraremos un esquema general del sistema de instrumentación.

5.2. Esquema

A continuación se va a describir el esquema hardware necesario para poder llevar a cabo el ensayo, aunque no es objeto de este trabajo fin de Master, se considera necesaria una breve descripción del hardware para una mejor comprensión del software.

Para una mejor comprensión se han etiquetado 3 posibles estados de funcionamiento; reposo, stress y medida.

La siguiente figura nos muestra el esquema general de funcionamiento, compuesto por la célula, en este caso representada con un diodo, la cual estará conectada a sendos relés y al conmutador Keitlley 3706. Para poder llevar a cabo las medidas, con los relés seleccionaremos en cada momento si deseamos que nuestra célula se encuentre en el estado de reposo, de estrés y de medida. Además podemos observar la fuente Agilent 6631B que inyectará corriente a partir de una tensión establecida para poder llevar a cabo las medidas.

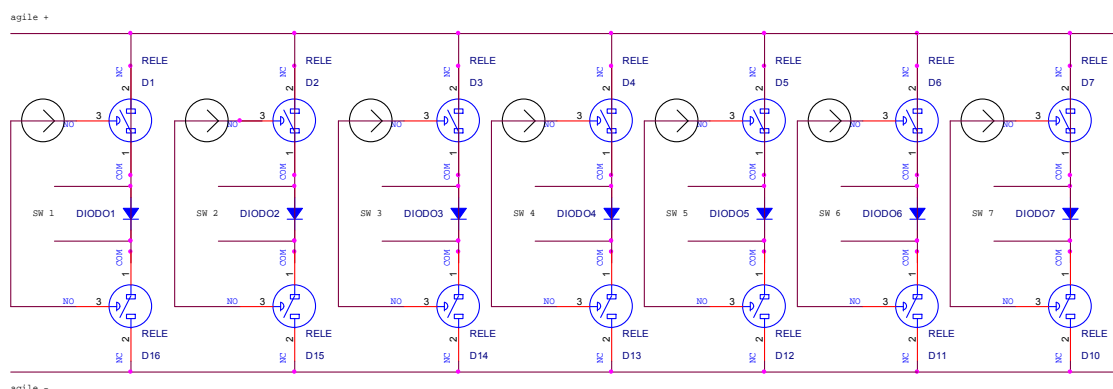


Figura 29 : Esquema general de conexionado

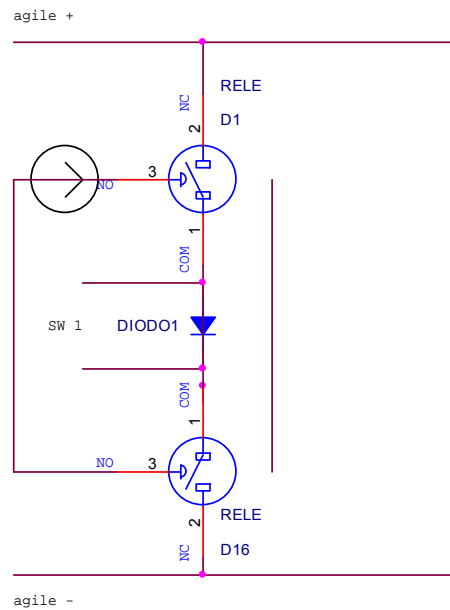


Figura 30 : Detalle de conexionado para 1 célula.

Además como medida de protección y para asegurar el estado de reposo se lleva a cabo un control de la alimentación mediante dos relés, de tal forma que si hubiera algún pico de corriente y el equipo se reiniciara las fuentes se apagarían, porque para poder estar conectada necesitará que un relé esté en la posición NO y el otro en la posición NC.

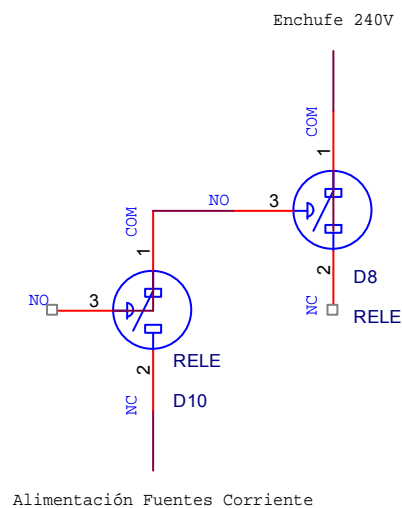


Figura 31 : Esquema de conexionado alimentación

A continuación vamos a describir los tres estados del sistema de instrumentación.

5.2.1. Estado de reposo

En el estado de reposo, como su propio nombre indica el circuito se encontrará sin ser sometido a ningún tipo de alimentación, ni de estrés, ni de medida. Dicho estado se utilizará como estado inicial del programa y además como estado intermedio desde que dejamos de someter a estrés a la célula hasta que comenzamos a tomar medidas para permitir a la célula estabilizarse en temperatura y que nuestras medidas no se vean influenciadas por la corriente a la que ha sido sometida en el estado de stress.

En este estado colocaremos todos los Relés en la posición NO, posición 1COM – 3NO de tal forma que tendremos un circuito cerrado como el que muestra el detalle de la figura del estado de reposo.

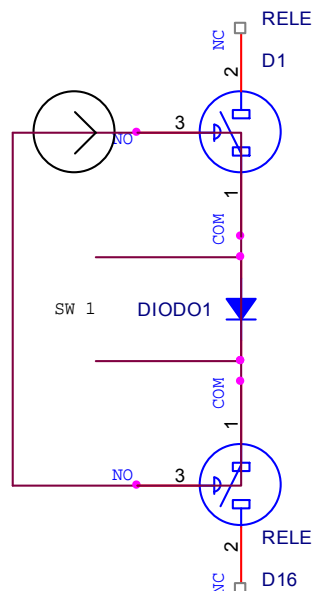


Figura 32 : Detalle estado reposo

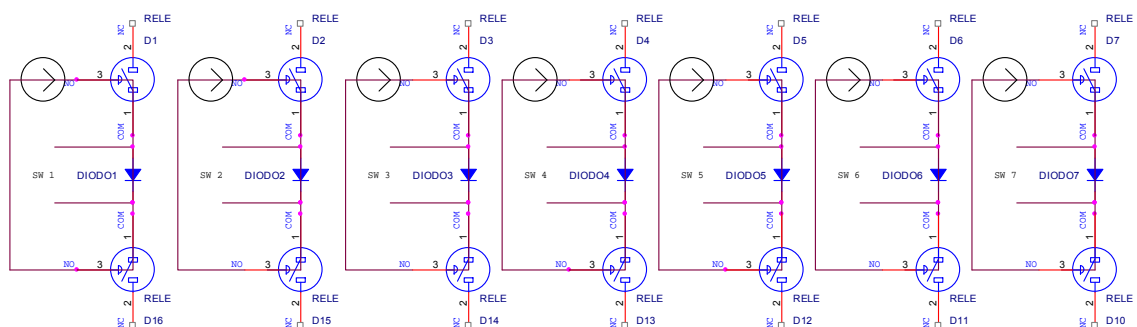
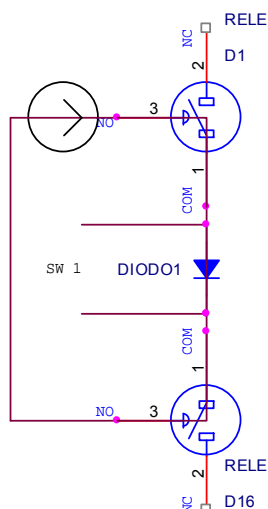
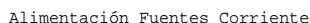


Figura 33 : Estado de reposo



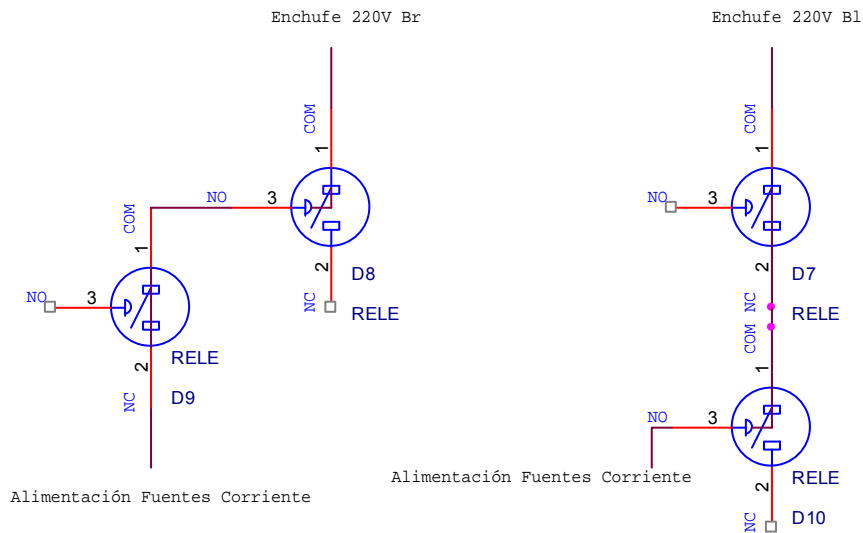


Figura 36 : Alimentación fuente de corriente

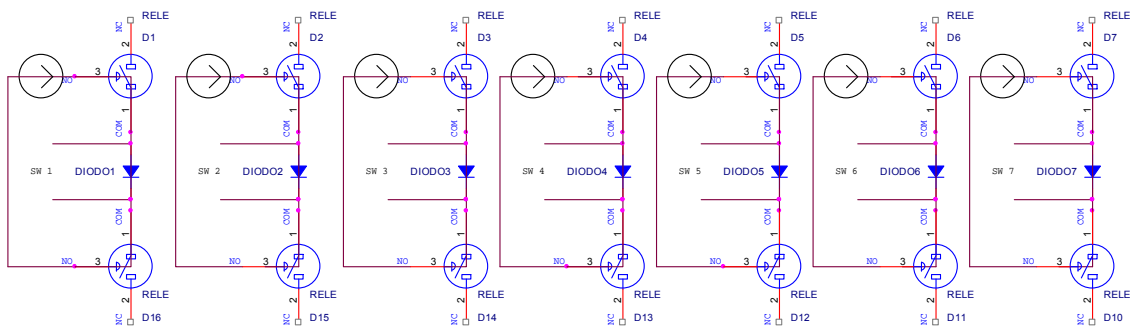


Figura 37 : Esquema circuito posición stress

5.2.3. Estado Medida

En el estado de medida se apagará la fuente de corriente, se colocan todos los relés en posición estrés, todos a NO, 1COM-3NO. Una vez hecho esto, se va seleccionando una a una la célula que queremos medir colocándola en posición NC los dos relés que rodean la célula en cuestión, 1COM-2NC, la denominada posición medida, en la figura siguiente corresponderían al relé 1 y al relé 16. Y una vez medido pasamos de nuevo a la posición STRESS 1COM-3NO.

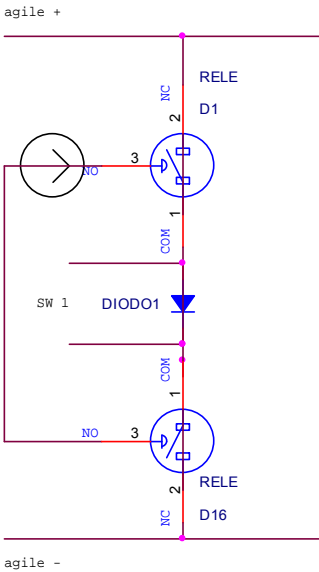


Figura 38 : Ejemplo Posición Medida célula 1

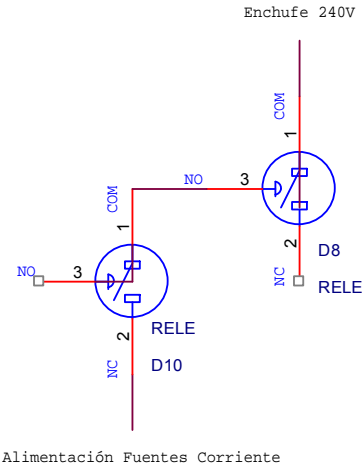


Figura 39: Posición alimentación ejemplo medida célula 1

Una vez descrito el esquema de los tres estados de funcionamiento pasaremos a explicar el esquema hardware implementado.

6 Esquema Hardware

6.1. Descripción del Hardware

Aunque el diseño del hardware no es objeto de este trabajo fin de Máster, se considera necesario una breve descripción del hardware usado para una correcta comprensión tanto del software como del funcionamiento general del ensayo acelerado. Si desea mayor información consulte el TFM de Eva Belmonte (Belmonte). A continuación realizaremos una breve descripción de los diferentes elementos del hardware.

6.1.1. Ordenador

Será necesario un PC que disponga de la versión de Labview 9.0 instalada, y los drivers de las tarjetas que se describirán a continuación. Así mismo los requisitos que se necesitan del PC son dos puertos USB 2.0, y conexión a Internet para poder realizar un control remoto de toda la instalación sin necesidad de estar presente físicamente. Además se recomienda que el procesador sea al menos un Pentium 4/M o su equivalente, RAM 1 GB, la resolución de la pantalla 1024 x 768 pixeles, con el Sistema Operativo Windows 7/Vista/XP. Espacio en Disco 3,3 GB.



Figura 40: Ordenador.

6.1.2. GPIB

Este dispositivo hardware se procede a describir con mayor detalle debida a su influencia software en el sistema.

1.1.1. Descripción de GPIB

GPIB (General Purpose Interface Bus) es un estándar de conexión que permite la comunicación de un ordenador con instrumentos electrónicos de medida, como pueden ser medidores de impedancias, multímetros, osciloscopios, etc.

Fue creado en 1965 por la compañía Hewlett-Packard, que lo denominó originalmente HP-IB (Hewlett-Packard Instrument Bus), fue creado para controlar los instrumentos programables que ellos mismos fabricaban y se popularizó con rapidez, debido a sus altas tasas de transferencia de datos (8 Mbytes/s). Para evitar la dispersión de características, en 1975, el Institute of Electrical and Electronic Engineers (IEEE) publicó el estándar ANSI/IEEE 488-1975, IEEE Interfaz Digital Estándar para Instrumentación Programable, que contiene las especificaciones eléctricas, técnicas y funcionales para la interconexión de sistemas.

Una segunda estandarización (IEEE 488.2 de 1987) delimitó de forma más concreta la programación del GPIB, definiendo comandos de aparatos, formato de mensajes y estado de los instrumentos.

Debido a que el documento original de la norma IEEE 488 no contenía directrices sobre sintaxis y formato de las instrucciones, se continuó trabajando para garantizar la compatibilidad y correcto funcionamiento entre los diferentes equipos de medida. Esto dio como resultado la norma IEEE 488.2, que contiene códigos, formatos, protocolos y comandos comunes para el bus IEEE 488 (renombrado entonces como IEEE 488.1). La nueva norma IEEE 488.2 no reemplaza a la anterior IEEE 488.1, de forma que muchos dispositivos pueden adaptarse sólo a la IEEE 488.1.

La IEEE 488.2 define unas características mínimas de la interfaz, un repertorio común de instrucciones, un protocolo de mensajes y un nuevo modelo de informe de estado.

En 1990, la norma IEEE 488.2 incluyó la norma SCPI (Standard Commands for Programmable Instrumentation), que define comandos específicos para cada tipo de instrumento y que incluyen todos los fabricantes. SCPI garantiza total compatibilidad entre instrumentos. En los sistemas SCPI es posible cambiar un instrumento por otro equivalente de otro fabricante con total compatibilidad.

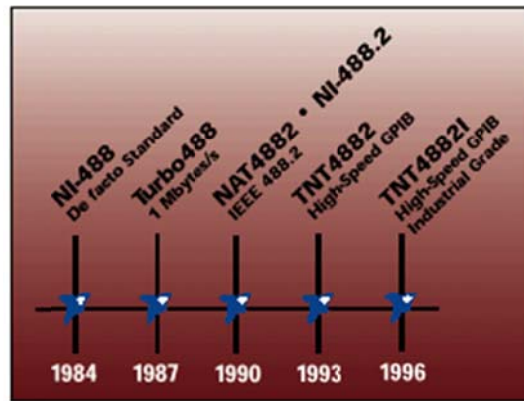


Figura 41 : Evolución GPIB

El principal objetivo del bus GPIB consiste en gestionar la transferencia de información entre dos o más dispositivos. Antes de enviar los datos hacia los dispositivos (instrumentos conectados al bus) éstos deben configurarse de acuerdo con este protocolo de transmisión de información. Entre los parámetros relativos al protocolo se encuentra la asignación de direcciones a los instrumentos interconectados.

La numeración del dispositivo, o asignación de su dirección, se realiza desde el panel frontal o alterando la conexión de los puentes de su tarjeta interfaz, que suele ser accesible desde la parte posterior del instrumento.

El elemento controlador del equipo GPIB es único (generalmente la tarjeta controladora instalada en un PC, en cuyo caso se le asigna la dirección 0), supervisa todas las operaciones que se realizan en el bus, y determina el dispositivo que envía la información y el momento en que se realiza su envío. El controlador puede designar un sustituto si en un determinado momento no puede atender los requisitos de control. El nuevo controlador recibe el nombre de controlador activo.

El controlador asegura que no puede haber dos o más instrumentos enviando información al bus simultáneamente. Además, establece los dispositivos que permanecen en estado de recepción o escucha, ya que no todos los instrumentos están siempre interesados en captar la información del bus. Esta función la realiza “despertando” a los dispositivos en estado de “latencia” mediante una solicitud de reafirmación, y mediante órdenes que especifican los nuevos receptores y el nuevo emisor.

Cuando el proceso de transmisión-recepción ha finalizado, el controlador del equipo se asegura de que todos los receptores han recibido la información enviada al

bus por el emisor mediante el “data hardware handshake” o control de transferencia de datos. Este protocolo permite asegurar la recepción de la información por parte de los dispositivos más lentos. Como consecuencia, el dispositivo más lento limita la velocidad de operación del equipo GPIB.

En resumen, se consideraran los siguientes elementos o conceptos más relevantes y específicos, involucrados en un equipo red de instrumentación mediante el protocolo GPIB:

- Controlador del equipo controlador activo.
- Dispositivos conectados al bus.
- Dispositivo fuente.
- Dispositivos destino.
- Comandos y funciones.

1.1.2. Estructura de un sistema GPIB

Un sistema GPIB consiste en una serie de instrumentos de medida conectados a un bus, y controlados, normalmente, por un PC dotado de una tarjeta GPIB, o un cable USB-GPIB. Existe bastante libertad en la configuración topológica del bus, que, en general, es una combinación de disposiciones lineales y en estrella, como se muestra en la figura siguiente.

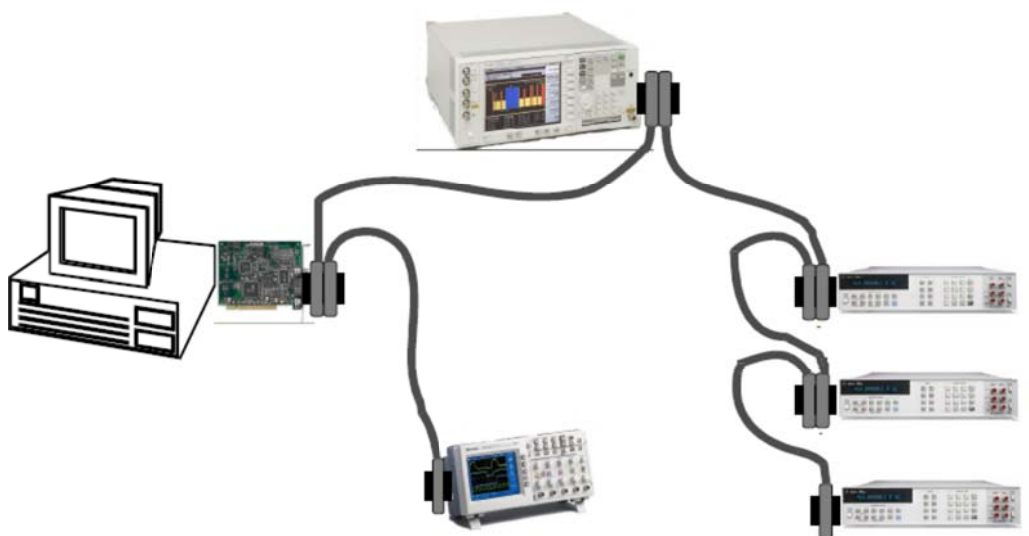


Figura 42 : Esquema conexionado GPIB

El dispositivo controlador (normalmente un PC dotado de una tarjeta GPIB) gestiona el flujo de datos y comandos a los diferentes elementos del sistema, en nuestro caso mediante la utilización del software Labview 9.0.

1.1.3. Características eléctricas del GPIB

El bus de transmisión de datos de GPIB es de 8 bits en paralelo, y lógica negativa con niveles TTL estándar (cierto si el voltaje es ≤ 0.8 V y falso si el voltaje es ≥ 2.0 V). Los cables y conectores tienen el aspecto típico mostrado en la figura siguiente:



Figura 43 : Cable de conexión GPIB: aspecto físico.

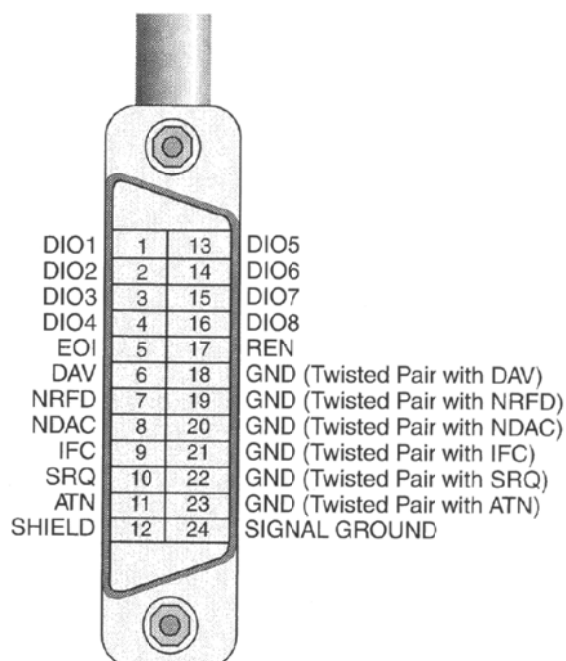


Figura 44: Cable de conexión GPIB: distribución de señales

La interfaz GPIB consta de 16 líneas y las 8 restantes del bus (hasta 24) corresponden a líneas de retorno a tierra. De las 16 líneas, 8 son de datos (1 byte) y 8

para mensajes de control y estados de los dispositivos. De estas últimas 8 líneas, 3 son para el control de transferencia de datos (handshake (recepción de cada byte)) y 5 para el control general de la interfaz.

1.1.4. Líneas de datos

Las 8 líneas de datos DIO1-DIO8 pueden transportar tanto datos como órdenes. El estado de la línea ATN (una de las 5 de control general) determina si son datos u órdenes los presentes en el bus. Si está a nivel bajo son órdenes o direcciones, y si está a nivel alto son datos. Todas las órdenes y la mayoría de los datos emplean 7 bits codificados en ASCII o ISO. En este caso el octavo bit se emplea para paridad o no se emplea.

1.1.5. Líneas de control de transferencia de datos (handshake)

Estas 3 líneas realizan el control asíncrono de las transferencias de los mensajes en forma de byte entre los dispositivos. Este proceso garantiza que la transmisión y la recepción se han realizado sin errores, dotando a la transmisión de información de seguridad. Las líneas son:

- NRFD (Not Ready For Data): Indica cuándo un dispositivo está preparado para recibir un byte. La línea es conducida por todos los dispositivos cuando reciben órdenes (cuando actúan como receptores de datos manteniéndose a la escucha), y por el orador o transmisor de datos cuando habilita el protocolo HS488.
- NDAC (Not Data Accepted): Indica cuándo un dispositivo ha aceptado un mensaje (en forma de byte). La línea es conducida por todos los dispositivos al recibir órdenes y/o datos, cuando reciben la información.
- DAV (Data Valid): Indica cuándo las señales en las líneas de datos se han estabilizado (se consideran válidas) y pueden ser aceptadas con seguridad por los dispositivos. El controlador conduce la línea de datos al enviar órdenes. Los transmisores de datos la conducen cuando envían mensajes de datos.

1.1.6. Líneas de control general de la interfaz

Las líneas de control general de la interfaz son las siguientes:

- SRQ (Service ReQuest): Se emplea para solicitar turno al controlador.
- IFC (InterFace Clear): Realiza un “reset” de los parámetros y direcciones del bus.
- REN (Remote ENable): Establece el control remoto de un dispositivo deshabilitando su control a través del panel. Es decir, el usuario ya no podrá realizar un control manual.
- EOI (End Or Identify): Se emplea como indicador de fin de transferencia de datos.
- ATN (ATtention): La emplea el controlador para distinguir entre los datos y los mensajes de control. Determina el tipo de información presente en las líneas de datos.

1.1.7. Especificaciones mecánicas

Cada dispositivo del equipo GPIB posee una dirección, codificada como un número entre 0 y 30. Existen 31 direcciones primarias permitidas para instrumentos transmisores de datos (“talkers”) conectados al bus interfaz GPIB. A cada instrumento se le asigna una dirección codificada mediante 7 bits. Puede haber hasta 15 dispositivos conectados en un bus contiguo, siendo sólo uno de ellos el controlador. Es decir, a una tarjeta controladora pueden conectarse hasta 14 dispositivos, por ejemplo encadenando cables IEEE-488 de un dispositivo al siguiente. Algunos dispositivos conectados al bus pueden direccionarse también mediante direcciones secundarias. Éstas hacen referencia a alguno de sus bloques funcionales. De los 15 dispositivos conectados no puede haber más de diez, 2/3 de los 15, funcionando simultáneamente.

La normativa IEEE - 488 especifica que pueden usarse hasta 20 metros de cable encadenado en línea, contando las conexiones; también recomienda no emplear tramos de cable de más de 4 metros de longitud. La alternativa de conexión en estrella se emplea para aumentar el número de dispositivos interconectados, pero tiene como inconveniente la mayor capacidad parásita de la red, que introduce retardos y, en consecuencia, errores de transmisión.

La velocidad de la señal es de 1 Mbyte por segundo en distancias cortas pero de forma general es usual es obtener velocidades comprendidas entre 250 y 500 kilobytes/segundo.

1.1.8. Funcionamiento. Transferencia de datos y cronograma

El estándar IEEE 488.1 establece un “handshake” de 3 líneas de control de datos. La introducción de esta lógica negativa permite la implementación de la función lógica OR mediante cableado. De esta forma, no se transmiten datos hasta que no esté listo el receptor (listener) más lento, y queda asegurado que la transmisión sea lo suficientemente lenta como para que al receptor más lento le dé tiempo a aceptar el dato.

La línea NRFD es controlada por cada receptor e indica si cada uno de ellos no está listo (nivel bajo) o lo está (nivel alto) para recibir datos. La línea DAV es controlada por el transmisor e indica si los datos en las líneas de datos (DIO) son correctos y, en consecuencia, pueden ser aceptados por los receptores. Finalmente, la línea NDAC es controlada por cada receptor para indicar que no ha recibido los datos (nivel bajo) o que los ha recibido (nivel alto).

La *Figura 45: Diagrama de tiempos de operación.* muestra el diagrama de tiempos de operación. En principio, el transmisor comprueba que las líneas NRFD (Not Ready For Data) y NDAC están a nivel bajo. La primera indica que no todos los receptores están listos para recibir datos y la segunda indica que no han aceptado ningún nuevo byte. Observar que la línea NRFD no pasa a nivel alto hasta que todos los receptores están listos. Una vez que el transmisor ha detectado que la línea NRFD está a nivel alto y transcurre cierto retardo, necesario para dar tiempo a estabilizar los niveles de los datos que envía a los receptores, pone la línea DAV a nivel bajo indicando que los datos que envía son válidos (instante 3). Se transfiere así un byte de datos.

El receptor más rápido pone la línea NRFD a nivel bajo con el fin de indicar que no está listo para recibir otro byte (instante 4). Los demás harán lo mismo cada uno a su ritmo. Es decir, el receptor más rápido indica al equipo que no mande más información porque él ha tomado ya la que había y tiene que aceptarla o procesarla (es posible que se requiera de él una respuesta). Finalmente, los receptores van aceptando el byte poniendo a nivel alto sus líneas NDAC. Cuando todos han aceptado los datos (instante 6), la línea pasa a nivel alto, el transmisor lo detecta y pone la línea DAV a nivel alto para indicar que ya no valen los datos (instante 7). El primer receptor que detecta que la línea DAV ha pasado a nivel alto pone la línea NDAC a nivel bajo (instante 8). El transmisor pondrá otros datos nuevos en las líneas DIO y comienza

otro nuevo ciclo. Un byte puede no transferirse en menos tiempo de la duración de los siguientes eventos:

- Propagación del cambio de estado de la línea NRFD hacia el emisor de datos (talker).
- Propagación de la señal DAV a todos los receptores.
- Que los receptores acepten el byte y confirmen la línea NDAC (la pasen a nivel alto).
- Que la señal NDAC regrese al emisor.
- Que el emisor permita el tiempo de establecimiento de los datos en el bus antes de confirmar la línea DAV (pasarla a nivel bajo).

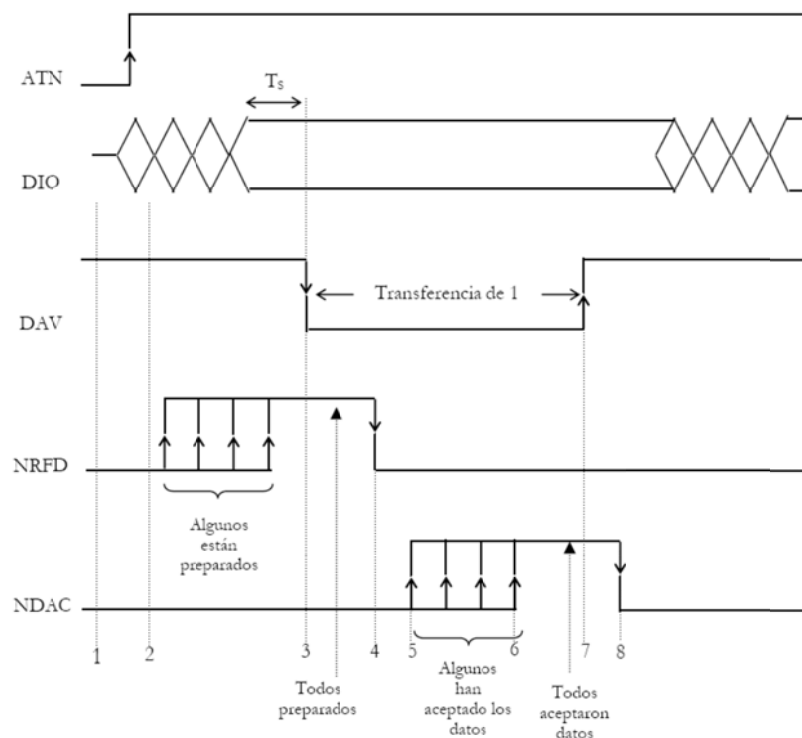


Figura 45: Diagrama de tiempos de operación.

1.1.9. Controlador GPIB HS USB

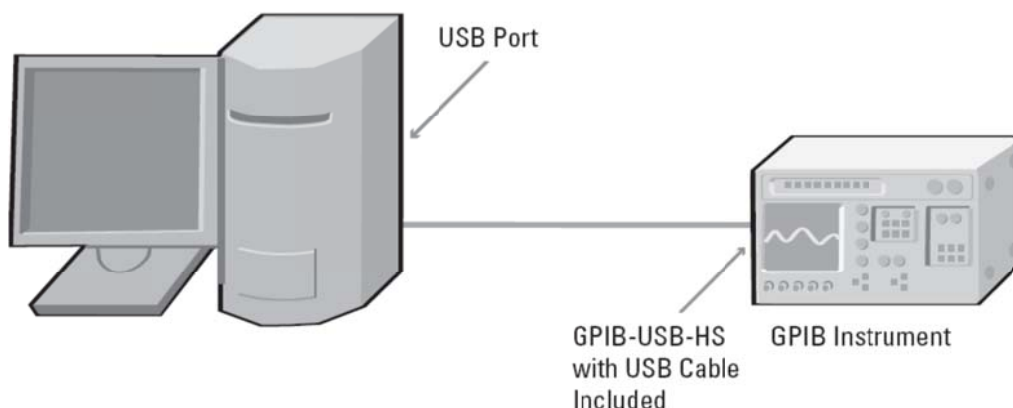


Figura 46 : Conexión Gpib usb

El controlador GPIB para USB 2.0 de alta velocidad (también es compatible con puertos 1.x) de National Instruments, es compacto y transforma cualquier ordenador con un puerto USB en un controlador completo IEEE 488.2 plug-and-play (enchufar y usar) para hasta 14 instrumentos GPIB programables. NI GPIB-USBHS aprovecha el USB 2.0 de alta velocidad para brindar un rendimiento superior, de hasta 1.8 MB/s con el protocolo IEEE 488 estándar y 7.2 MB/s con el protocolo IEEE 488 de alta velocidad (HS488). Por ser pequeño y ligero, es ideal para aplicaciones portátiles usando un ordenador portátil u otras aplicaciones donde el ordenador no tiene ranuras internas de E/S.

De ésta manera, no se requiere de un cable GPIB para la conexión de un instrumento. En el caso de necesitar conectar varios instrumentos, éstos se conectarán mediante sus respectivos cables GPIB y se conectará el GPIB-USB-HS en la terminación final, de manera que se queden todos los instrumentos conectados al ordenador a través del puerto USB de la misma manera que se haría a través de una tarjeta GPIB.



Figura 47: Controlador GPIB USB HS

6.1.3. Relés OMROM G5LA

El relé o relevador es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.



Figura 48: Omrom G5LA

Las principales especificaciones eléctricas de estos relés son:

- Máxima potencia conmutación 10A 250VAC
- Soporta hasta 4,500V
- Consumo de la bobina:360mW
- Contactos libres de Cadmio

Especificaciones mecánicas:

Carga nominal	10A at 250VAC (N.O) 10A at 24VDC (N.O) 5A at 125VAC (N.O/N.C) 5A at 24VDC (N.O/N.C)
Corriente nominal	10A(N.O) 5A(N.C)
Max. Tensión de conmutación	250VAC 24VDC
Max. Corriente de conmutación	AC10A DC10A(N.O) AC5A DC5A (N.C)
Max. alimentación	AC2500VA DC240W(N.O) AC625VA DC120W(N.C)
Referencia de fallo	100mA at DC5V

Tabla I: Especificaciones mecánicas relé Omrom G5LA

Características:

Resistencia de los contactos	100 m Ω max.
Tiempo de operación	10 ms max.
Tiempo de respuesta	5 ms max.
Max. Frecuencia de operación	Mechanical: 18,000 operations/hr Electrical: 1,800 operations/hr(under rated load)
Resistencia de aislamiento	1,000 M Ω min. (at 500 VDC)
Dieléctrico strength	2,000 VAC. 1mA 50/60Hz for 1 min between coil and contacts 750 VAC 1mA 50/60Hz for 1 min between contacts of same polarity
Resistencia a vibraciones	Destruction: 10 to 55 Hz, 1.5-mm double amplitude Malfunction: 10 to 55 Hz, 1.5-mm double amplitude
Resistibilidad	Destruction: 1,000 m/s ² (approx.100G) Malfunction: 100 m/s ² when energized ; 100 m/s ² when no energized
Durabilidad	Mechanical: 10,000,000 operations min. (at 18,000 operations/hr under no load) Electrical: 100,000 operations average. (at 1,800 operations/hr under rated load)
Temperatura ambiente	Operating: -40°C to 85°C (with no icing) Storage: -40°C to 85°C (with no icing)
Humedad Ambiente	Operating: 35% to 85% Storage: 35% to 85%
Peso	Approx. 7.5g

Tabla II: Características relé Omron G5LA

6.1.4. Tarjeta 16 relés

Usaremos la tarjeta relés con 16 relés OMROM G5LA. Sus principales características son:

- Alimentación 12 VDC.
- 160mm x 100mm.
- Relés aguantan 240V@ 10A en NO (Normalmente abiertos) y 5A en NC (Normalmente cerrado).
- Los 3 contactos del relé son accesibles (NO, COM and NC).
- LED indicador de la posición de los relés, ojo esto indica la señal digital que le ha llegado al relé para que este actúe en consecuencia, no es una comprobación del estado del relé.
- Señal de control de entrada digital compatible con lógica TTL (0 / 5 VDC).
- Señal de control de entrada digital mediante 2 conectores de 10 pines.
- Diodo de protección ante posible inversión de alimentación.

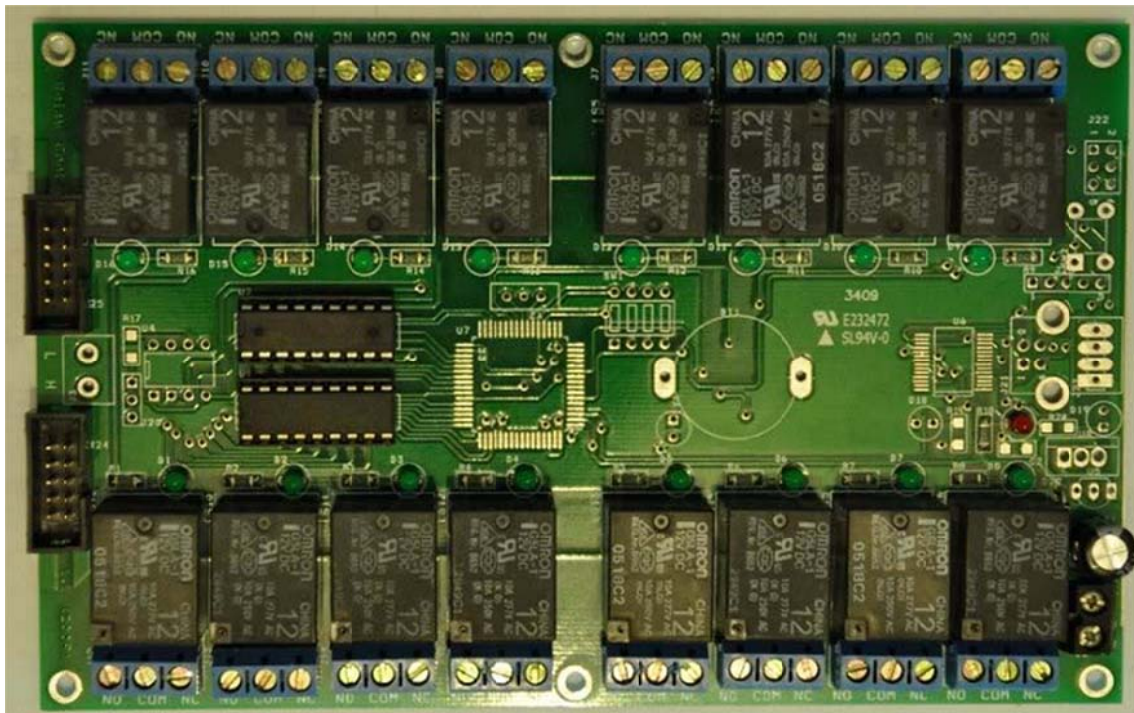


Figura 49: Tarjeta relés

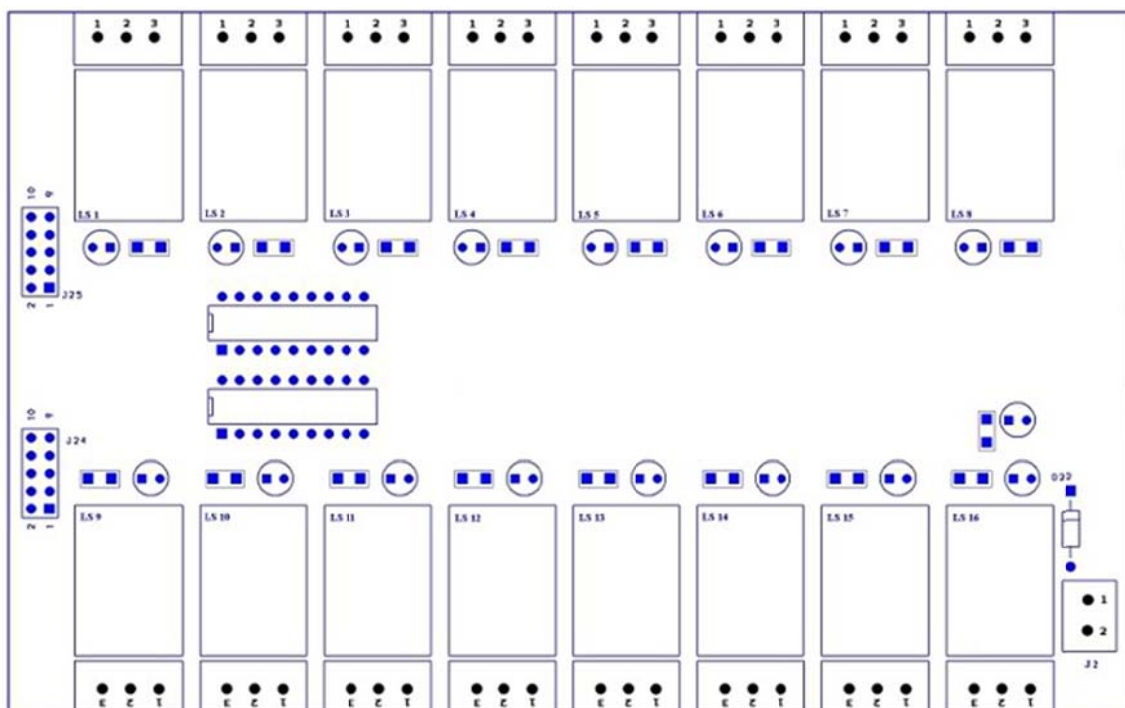


Figura 50: Esquema tarjeta relés

6.1.5. Tarjeta de adquisición de datos NI USB 6501

El USB-6501 de National Instruments es un dispositivo de E/S digital portátil, que brinda adquisición de datos y control confiables a un precio asequible. Con conectividad USB plug-and-play, el NI USB-6501 es lo suficientemente robusto y versátil para aplicaciones de laboratorio e industriales. Sus principales características son:

- 24 líneas de E/S digitales; un contador de 32 bits.
- Protección de sobre voltaje, 8.5mA de capacidad de corriente.
- Terminales de tornillo integradas o conector genérico de 34 pines (versión OEM) para fácil integración.
- Interfaz de bus USB 2.0 de alta velocidad (12 Mb/s).



Figura 51: NI USB 6501

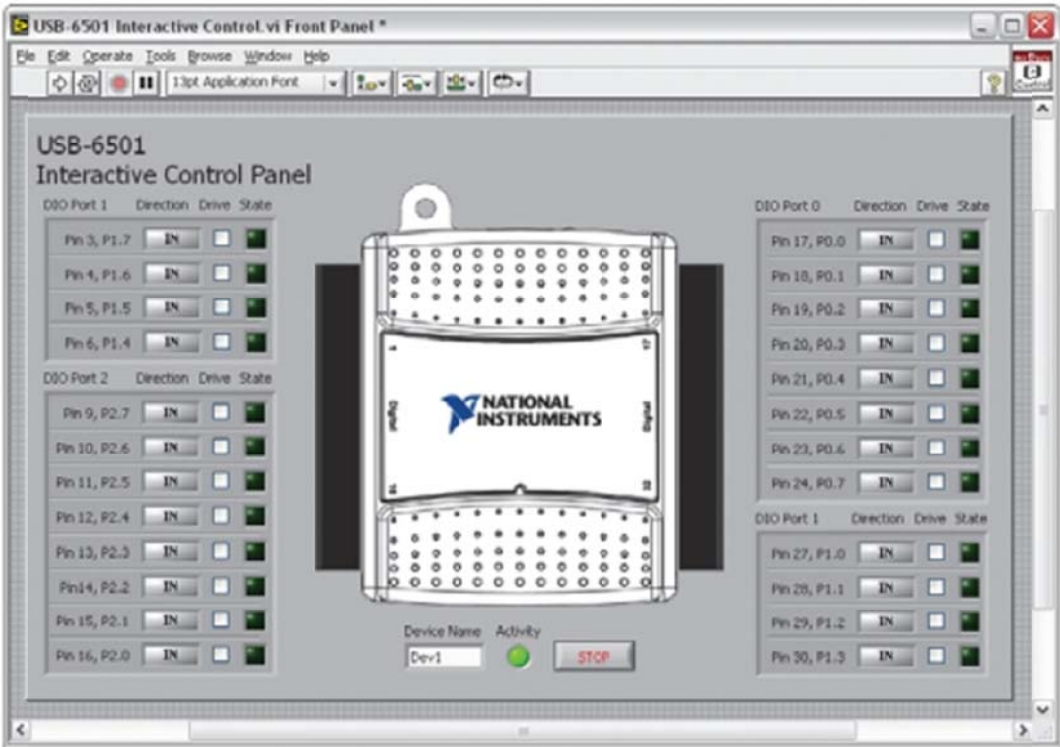


Figura 52: Labview NI 6501

Module	Terminal	Signal
	1	GND
	2	+5 V
	3	P1.7
	4	P1.6
	5	P1.5
	6	P1.4
	7	GND
	8	GND
	9	P2.7/PFI 0
	10	P2.6
	11	P2.5
	12	P2.4
	13	P2.3
	14	P2.2
	15	P2.1
	16	P2.0

Module	Terminal	Signal
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0.6
	24	P0.7
	25	GND
	26	GND
	27	P1.0
	28	P1.1
	29	P1.2
	30	P1.3
	31	+5 V
	32	GND

Figura 53: Diagrama pines Ni USB 6501

En la siguiente figura se muestra la implementación realizada de la interconexión entre la tarjeta de relés SPE 3721 con el DAQ NI ISB 6501, para poder controlar la conmutación de relés mediante la señal digital proveniente del DAQ gobernada por el software en Labview que se explicará en el siguiente capítulo.

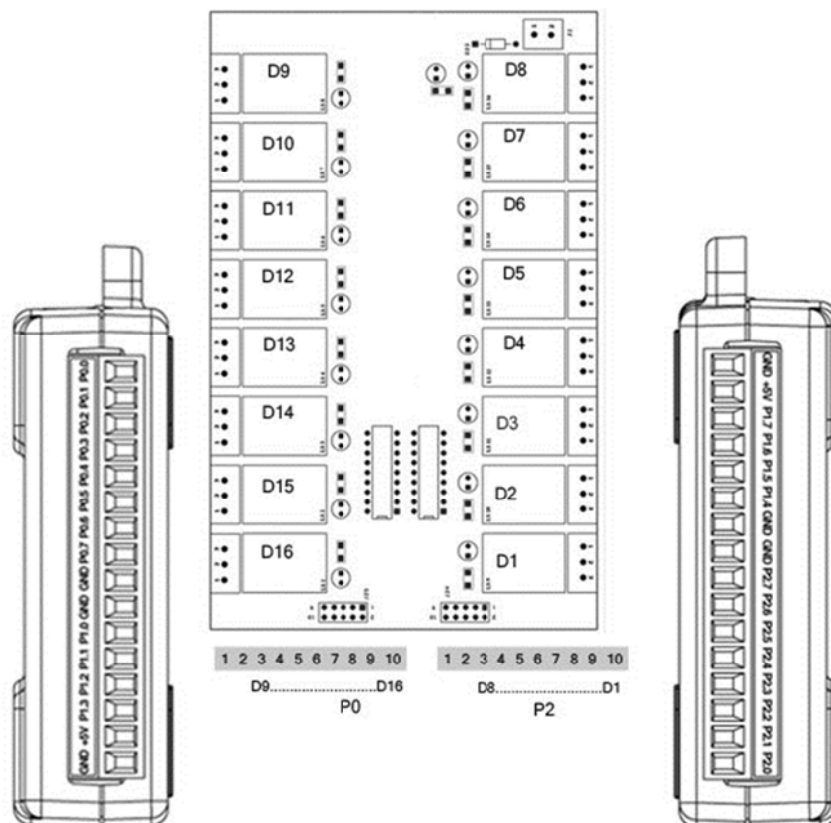


Figura 54: Interconexión con tarjeta relés

Equivalencia de NI USB 6501 con la tarjeta de Relés Omrom, según el esquema de interconexión de la figura anterior.

<i>PIN 17 P0.0</i>	<i>D9</i>
PIN 18 P0.1	D10
PIN 19 P0.2	D11
PIN 20 P0.3	D12
PIN 21 P0.4	D13
PIN 22 P0.5	D14
PIN 23 P0.6	D15
PIN 24 P0.7	D16

Tabla III: Equivalencia pines puerto 0

PIN 9 P2.7	D1
PIN 10 P2.6	D2
PIN 11 P2.5	D3
PIN 12 P2.4	D4
PIN 13 P2.3	D5
PIN 14 P2.2	D6
PIN 15 P2.1	D7
PIN 16 P2.0	D8

Tabla IV: Equivalencia pines puerto 2

6.1.6. Concentrador USB

Dado que el número de células a conectar son como máximo 40, es necesario conectar varios DAQ NI USB 6501, para ello, se optó con conectar un concentrador USB, de la marca Sweet, con 7 puertos Usb 2.0 con una velocidad de conexión de 480Mbps y alimentación externa.



Figura 55 : Concentrador USB

6.1.7. Fuente de Corriente 1A.

Según las especificaciones iniciales, la fuente de corriente necesaria para simular las condiciones de funcionamiento de la célula se estimaban en 1 Amperio, por lo que se comenzó con un primer diseño de una fuente de corriente basa en el siguiente esquema.

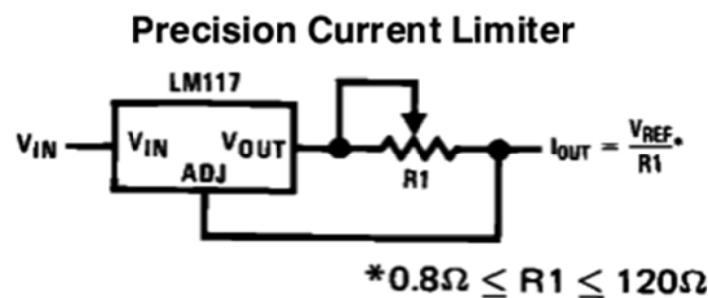


Figura 56 : Esquema Fuente Corriente 1 Amperio

Este circuito fija la corriente de salida, independientemente de la tensión que circule por el circuito mediante un potenciómetro, R1.

$$I_{out} = \frac{V_{ref}}{R1} \quad (71)$$

Para R1 entre 0,8 y 120Ω se puede limitar a corrientes entre 1,5A y 10mA. Si se usa una resistencia variable hay que tener cuidado de no pasarnos de los límites porque corremos riesgo de dañar la fuente o el circuito.

Una vez comprobado el correcto funcionamiento y la estabilidad de una de las fuentes de corriente de corriente, se pasó a montar 7 fuentes de corriente para realizar las pruebas de funcionamiento.

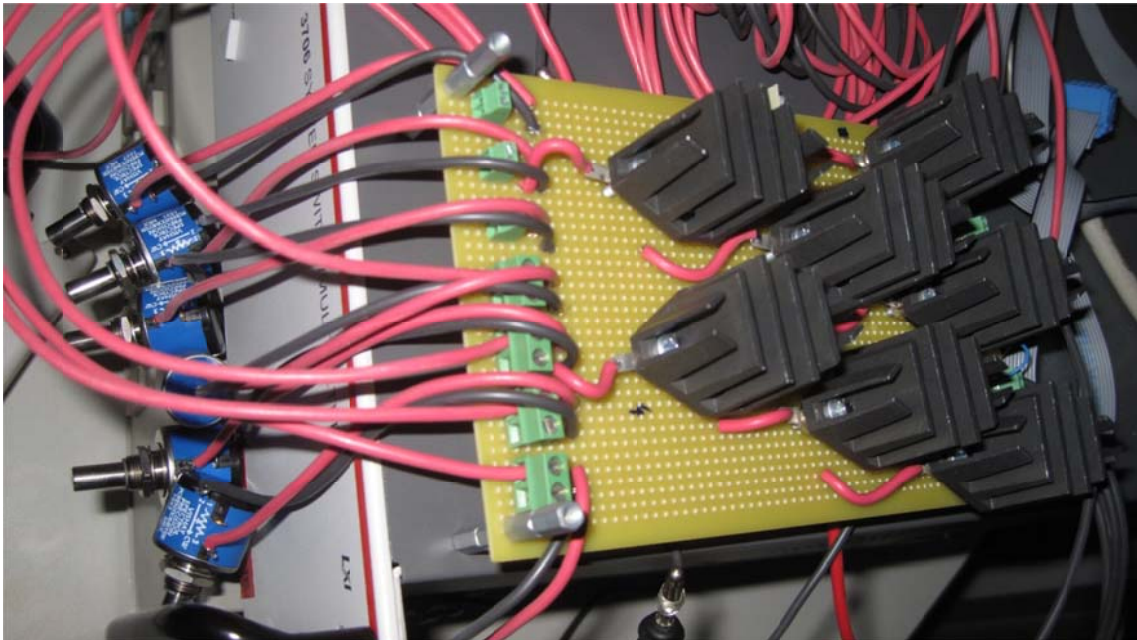


Figura 57: Montaje fuentes de 1A.



Figura 58: Primera simulación de conexionado.

A pesar del buen funcionamiento de las fuentes de corriente éstas se descartaron debido a un cambio de especificaciones en la corriente de emulación de las condiciones de funcionamiento por lo que se optó por una fuente de corriente comercial que será descrita a continuación.

6.1.8. Fuente de corriente ELN 60 3,2A

Tras las primeras pruebas de caracterización de la célula por parte del IES-UPM se decidió que la corriente a la que se le debía someter a la célula para emular las condiciones de funcionamiento tenía que ser de 3.2Amperios, por lo que se optó por el modelo ELN 60 incluyendo un potenciómetro para poder ajustar con precisión la corriente.



Figura 59: Fuente de corriente

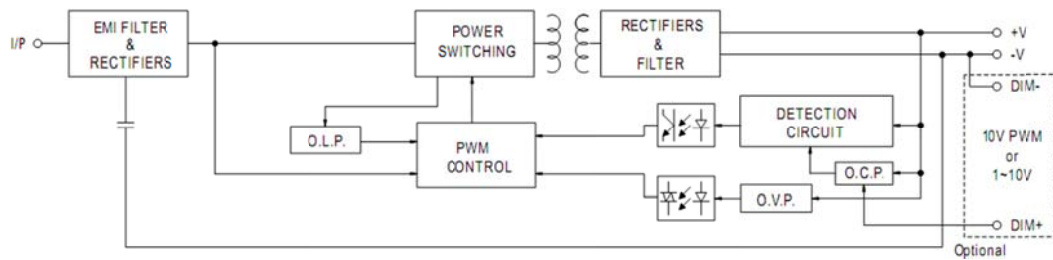


Figura 60: Circuito fuente de corriente



Figura 61: Interior de la fuente de corriente

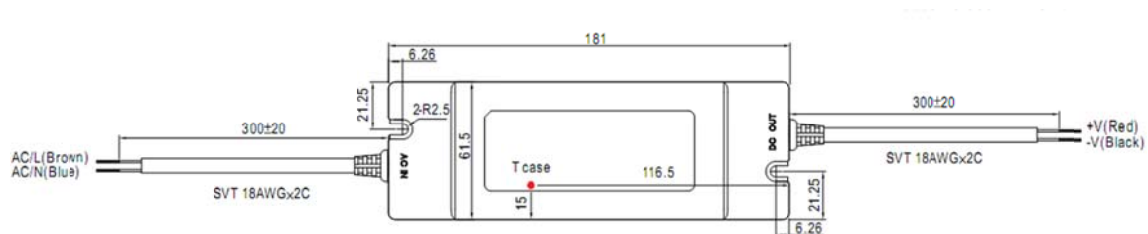


Figura 62: Dimensiones fuente de corriente.

6.1.9. Keithley 3706

El sistema de conmutación modelo 3706 con multímetro digital de alto rendimiento (DMM) contiene 6 ranuras para tarjetas conectables en una carcasa de 2U de alto (3.5 pulgadas/89mm) que se acomoda fácilmente a las necesidades de aplicaciones de media y alta densidad de canales. Cuando está totalmente conectado, puede soportar hasta 576 canales multiplexados de 2 hilos. El multímetro de alto rendimiento proporciona un sistema de conmutación y medición altamente integrado que satisface las exigencias de aplicaciones en un sistema de test funcional y entrega

la flexibilidad necesaria en aplicaciones autónomas de adquisición de datos y medición, como es el objeto de este proyecto.



Figura 63: Keithley 3706

Las principales características de este equipo son:

- Sistema de conmutación de 6 ranuras.
- Multímetro con 7 ½ dígitos con resolución de 10 nV; 1 pA ; 0,1μΩ, 14.000 lecturas/ segundo.
- Sistema LXI de clase B con control preciso de tiempo según protocolo IEEE1588.
- Incluye interfaces de comunicación LXI, ETHERNET, GPIB y USB 2.0.
- Arquitectura multi-procesador, TSP interno propio programable por el usuario que permite aplicaciones de control y medida complejas y autónomas, aumentando la velocidad de pruebas y simplificando las aplicaciones.
- Hasta 576 canales multiplexer de 2 alambres en un mainframe.
- Conformidad LXI Clase B con sincronización de tiempo IEEE 1588.
- Conexión master/slave TSP-Link™ que proporciona una fácil expansión del sistema y una conexión simple a los instrumentos SourceMeter® de la serie 2600 .Capaz de realizar alrededor de 14,000 lecturas por segundo con multímetro de alto rendimiento opcional.
- Interfaz de navegador web embebido para configuración de test, mantención y control básico de aplicación.

6.1.10. Tarjeta multiplexora 3721 Dual 1 × 20

Tarjeta multiplexora con 20 canales duales, con posibilidad de medida a 4 hilos, disponiendo de 20 canales a medir, o a 2 hilos, en dicho caso, se podrían usar hasta 40 canales.



Figura 64: Keithley 3721

Las principales características de esta tarjeta son:

- 40 canales conmutables para medidas a 2 hilos o 20 canales para medidas a 4 hilos.
- 2 canales dedicados para medir corriente hasta 3Amperios.
- Medida automática de la temperature con el accesorio 3721-ST
- Máxima señal soportada 300V, 2A switched or 3A carry. 60W, 125VA.
- Canales 41-42: 60V DC o 30V RMS, 3A switched, 60W, 125VA máximo. Fusible 3A, 250V RMS.
- Relés electromecánicos.

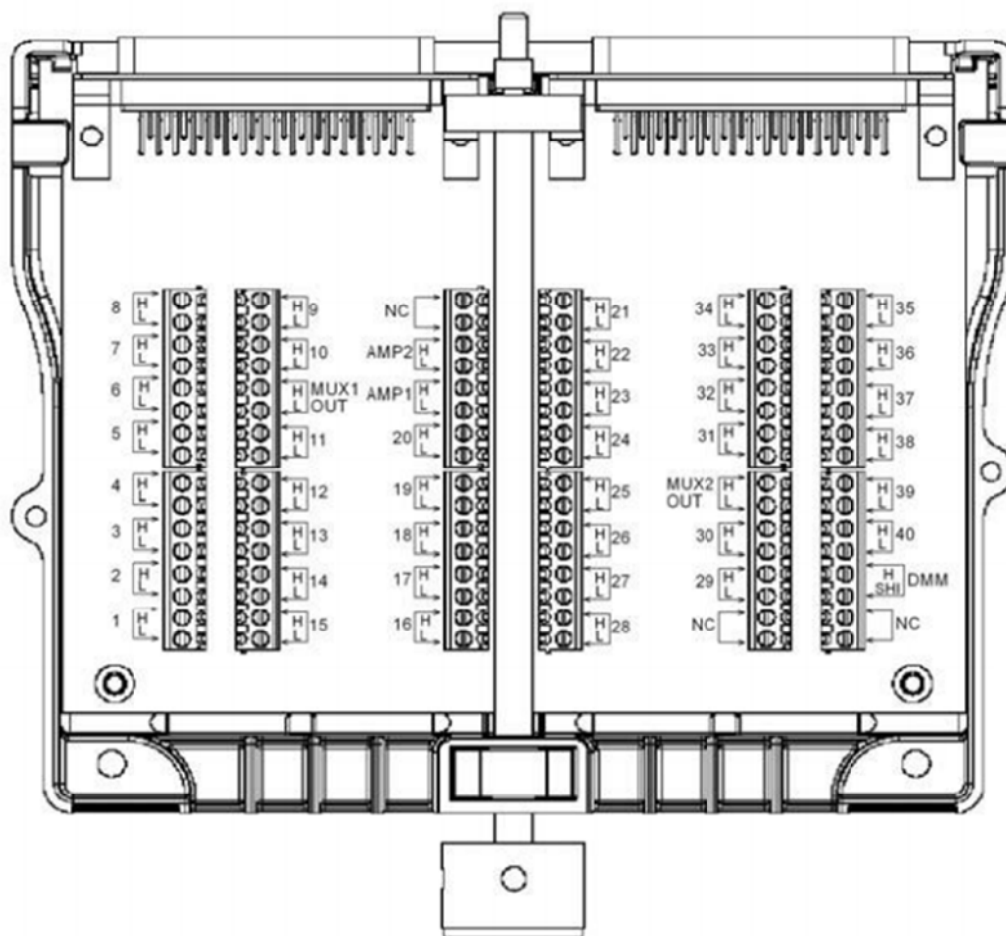


Figura 65 : Esquema pines tarjeta 3721

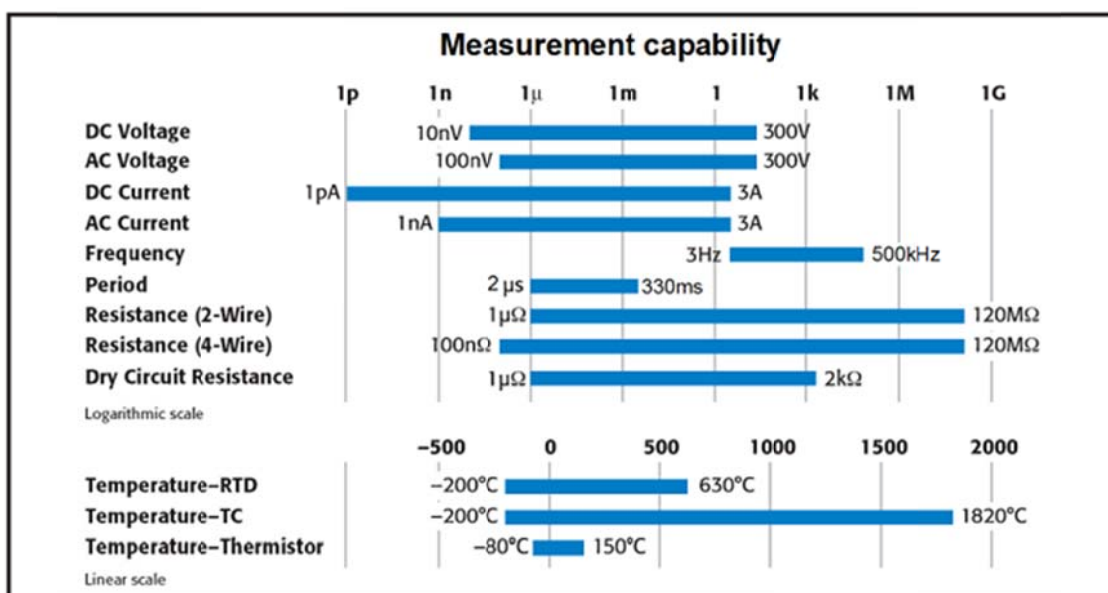
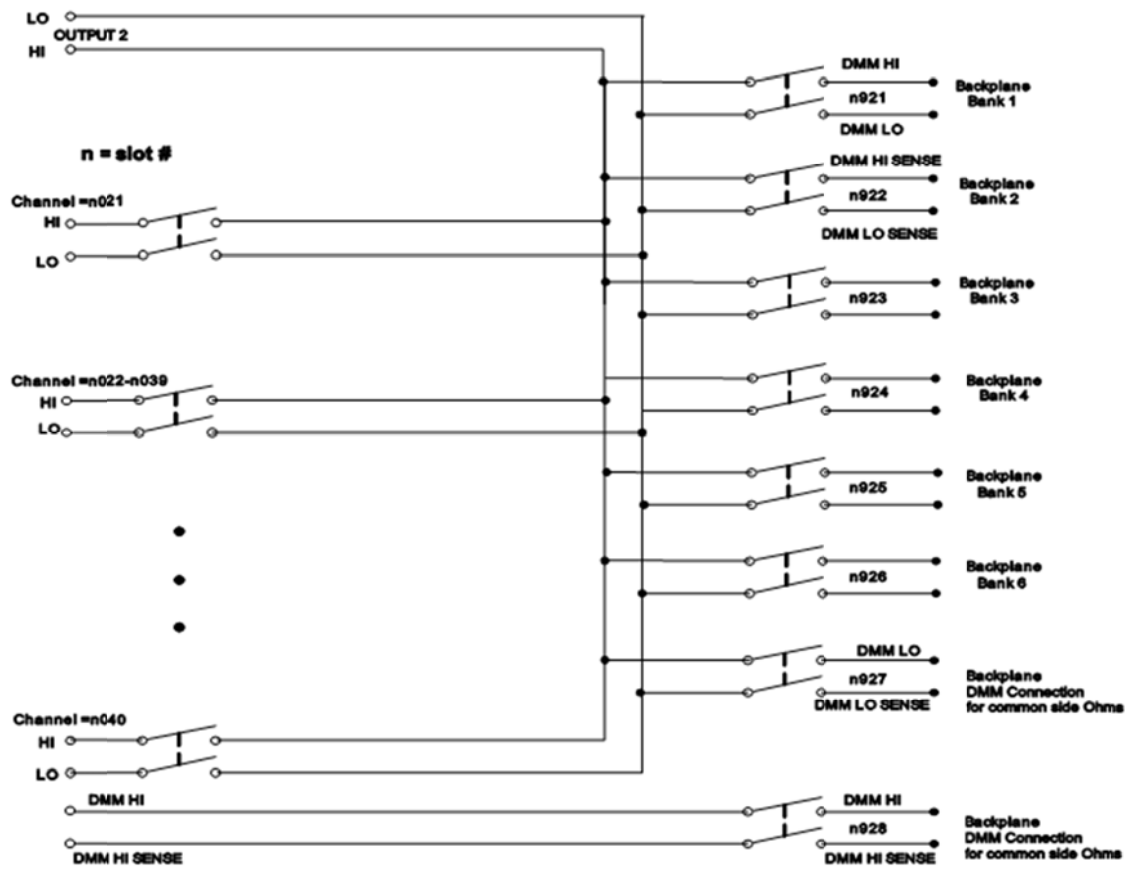
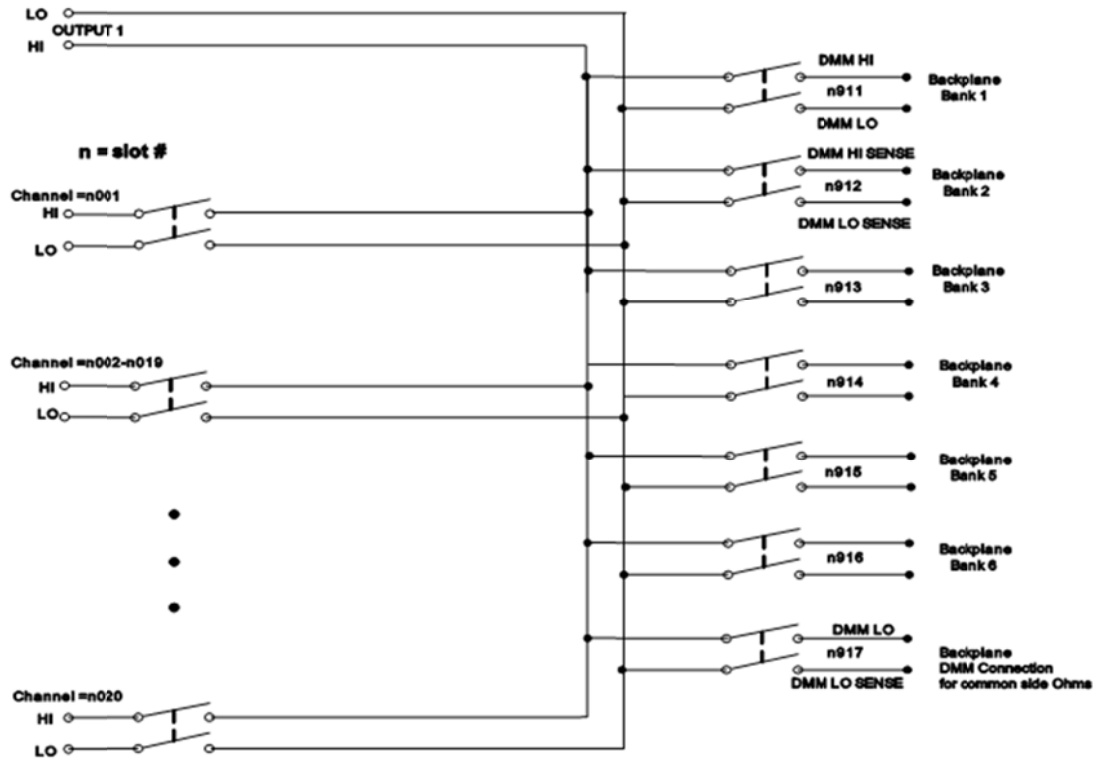


Figura 66: Capacidad de medida



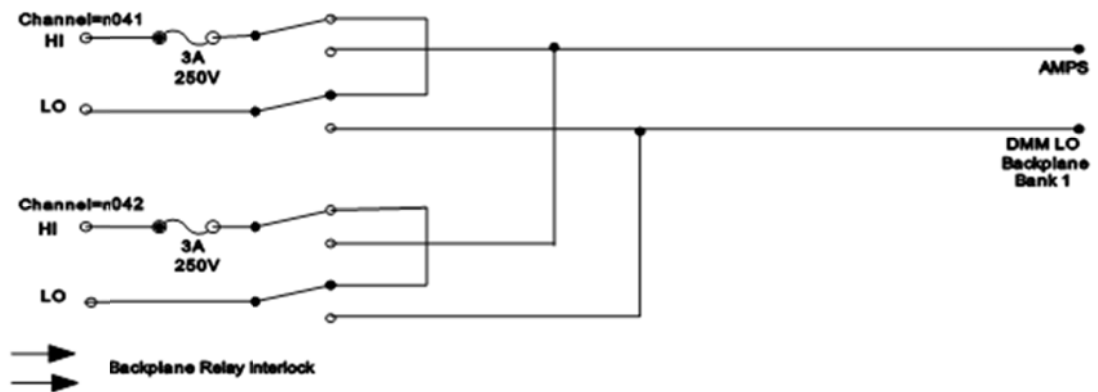


Figura 67: Conexión interno

6.1.11. Fuente de alimentación Agilent 6631B

Fuente de alimentación DC con capacidad de medida, ofrece rapidez y precisión, utilizada para caracterizar las células, de tal forma que introduce una tensión a la célula y mide la corriente que hay sobre la misma.



Figura 68: Agilent 6631B

Las principales características del equipo son:

- Puede ser programada remotamente vía GPIB o vía rs232, puerto serie.
- Tiene una salida, 80-100W.
- Voltage 0 – 8V.
- Corriente 0 10^aA.
- Precisión de medida a 25°C ±5°C

- Voltage: 5 mV
- Corriente 0.05% + 4 mA
- Ruido (20 Hz to 20 MHz)
 - Voltage normal modo rms: 0.3 mV
 - Voltage normal modo peak-to-peak: 3 mV
 - Fast mode rms/peak-to-peak: 1 mV/10 mV
 - Corriente rms: 3 mA

6.1.12. Cámara climática



Figura 69 Cámara climática

Se ha empleado una estufa universal de precisión modelo digitronic de la empresa JP Selecta. Estas cámaras están indicadas, entre otros usos, para ensayos acelerados en temperatura, que es nuestro caso.

Las características principales de esta cámara son:

- Circulación por aire forzado.
- Temperaturas regulables desde ambiente +5°C hasta 250°C.

- Estabilidad $\pm 0,25^{\circ}\text{C}$, a 100°C .
- Homogeneidad: $\pm 2\%$ a temperatura de trabajo.
- Error de consigna: $\pm 2\%$ a temperatura de trabajo.
- Resolución: 1°C .
- Regulador por microprocesador y lectura digital de temperatura y tiempo.

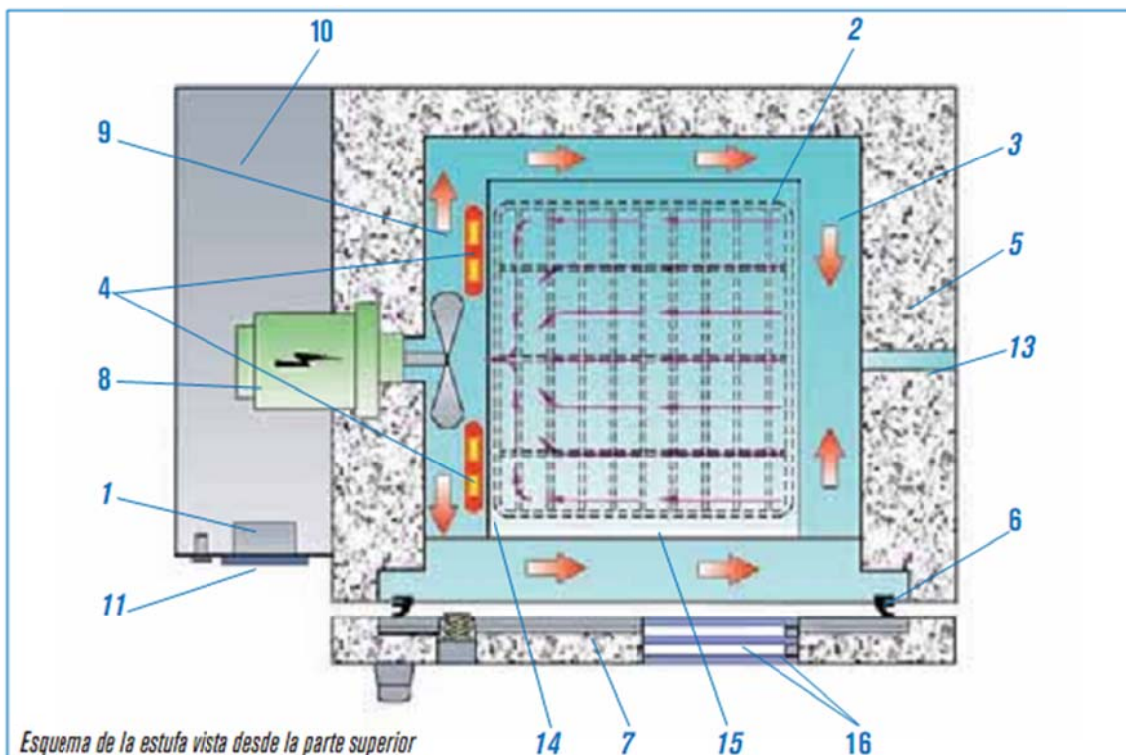


Figura 70: Esquema de la estufa

6.1.13. Sonda de temperatura.

Como sonda de temperatura se valoran y programan dos opciones, el termopar y la resistencia RTD (del inglés Resistance Temperature Detector).

Un termopar (también llamado termocupla) es un transductor formado por la unión de dos metales distintos que produce un voltaje (efecto Seebeck), que es función de la diferencia de temperatura entre uno de los extremos denominado "punto caliente" o unión caliente o de medida y el otro denominado "punto frío" o unión fría o de referencia.

En instrumentación industrial, los termopares son ampliamente usados como sensores de temperatura. Son económicos, intercambiables, tienen conectores estándar y son capaces de medir un amplio rango de temperaturas. Su principal limitación es la exactitud ya que los errores del sistema, inferiores a un grado Celsius, son difíciles de obtener.

En los ensayos se usará un Termopar tipo K (Cromo - Constantán) cuyas principales características son:

- Termoelemento positivo (KP): Ni(90%) Cr(10%).
- Termoelemento negativo (KN): Ni(95%) Mn(2%) Si(1%) Al(2%)
- Rango de utilización de -270° a 1200°.
- Puede utilizarse en atmósferas inertes y oxidables.
- No usar en atmósferas reductoras y sulfurosas.

La segunda sonda de temperatura utilizada, son las termoresistencias RTD, también conocidas como Pt100, es un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura.

Al calentarse un metal habrá una mayor agitación térmica, dispersándose más los electrones y reduciéndose su velocidad media, aumentando la resistencia. A mayor temperatura, mayor agitación, y mayor resistencia.

La variación de la resistencia puede ser expresada de manera polinómica:

$$R = R_0 \cdot (1 + \alpha \cdot \Delta T) \quad (72)$$

donde:

R_0 es la resistencia a la temperatura de referencia T_0

ΔT es la desviación de temperatura respecto a T_0 $\Delta T = T - T_0$

α es el coeficiente de temperatura del conductor especificado a 0 °C, interesa que sea de gran valor y constante con la temperatura,

El material seleccionado es platino, con un coeficiente de temperatura de 0.00385. Además con este material se aseguran medidas exactas y estables hasta 500°C.

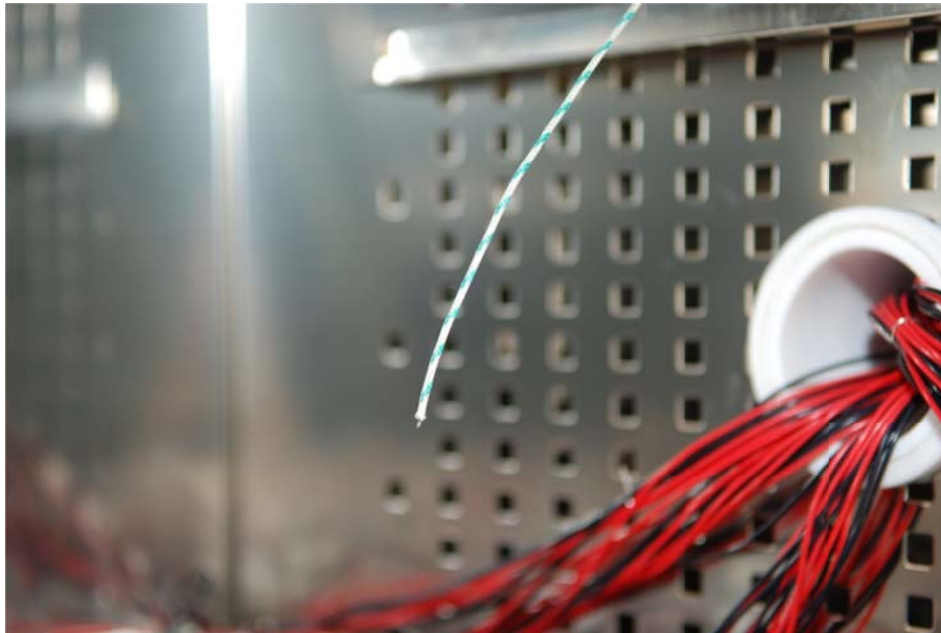


Figura 71: Pt100

6.1.14. Alimentación

Para la alimentación de los distintos dispositivos se necesitan alimentaciones con distintos voltajes:

- la primera de ellas 220VAC, la cual es necesaria para la conexión del ordenador y los equipos de medida, el Keithley 3706 y el Agilent 6631B. Así mismo, es necesaria para la alimentación de todas las fuentes de corriente, que están unidas a un mismo punto controlado por la tarjeta de relés para asegurar su funcionamiento.
- La segunda alimentación, de 5VDC, es necesaria para alimentar los ventiladores destinados a la refrigeración de las fuentes de corriente, y para la alimentación del concentrador USB.
- La tercera alimentación, de 12VDC, es necesaria para la alimentación de las tarjetas de relés.

6.1.15. Relés HF102F



Figura 72: Relé HF102F

Relé de potencia que resiste corrientes de entrada de hasta 80 A. Se han usado dos de estos relés para pasar la alimentación de 220V de la red a las fuentes de corriente ELN 60 3,2 A. Estos relés son gobernados por los relés de la tarjeta de 16 relés, y son usado para apagar o encender las fuentes de corriente cuando sean necesarias. Este relé se colocó en el diseño final dado que al encenderse las 20 fuentes, generaban un pico de corriente de 50 A, incluso empleando una resistencia de $4,7 \Omega$ en serie con los 220V, por lo que los relés Omron no lo soportaban.

7 Lenguaje de programación G

En este capítulo haremos una introducción al lenguaje de programación que se ha utilizado para controlar el sistema de instrumentación, en concreto el lenguaje de programación G, para implementarlo se ha usado una herramienta gráfica destinada para el diseño, pruebas y control denominada Labview.

7.1. LabView

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) de National Instruments es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control.

LabVIEW es una herramienta diseñada especialmente para monitorizar, controlar, automatizar y realizar cálculos complejos de señales analógicas y digitales capturadas a través de tarjetas de adquisición de datos, puertos serie y GPIBs. Es compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab.

Labview permite diseñar interfaces de usuario mediante una consola interactiva basada en software. Se puede diseñar especificando su sistema funcional, su diagrama de bloques o una notación de diseño de ingeniería.

Es un lenguaje de programación de propósito general, como es el Lenguaje C o Basic, pero con la característica de que es totalmente gráfico, facilitando de esta manera el entendimiento y manejo de dicho lenguaje para el diseñador y/o programador. Sus principales características son:

- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.

Incluye librerías para el manejo de:

- Interfaces de comunicaciones:
 - Puerto serie
 - Puerto paralelo
 - GPIB
 - PXI
 - VXI
 - TCP/IP, UDP, DataSocket
 - IrDA
 - Bluetooth
 - USB
 - OPC...
- Capacidad de interactuar con otras aplicaciones:
 - dll
 - ActiveX
 - Matlab
 - Simulink...
- Herramientas para el procesamiento digital de señales.
- Visualización y manejo de gráficas con datos dinámicos.
- Adquisición, análisis, presentación y almacenamiento de datos e imágenes.
- Control de movimiento.
- Tiempo Real estrictamente hablando.
- Programación de FPGAs.

- Sincronización.

Está basado en la programación modular, lo que permite crear tareas muy complicadas a partir de módulos o sub-módulos mucho más sencillos. Además, estos módulos pueden ser usados en otras tareas, con lo cual permite una programación más rápida y provechosa.

También ofrece la ventaja de depuración, “debugging” en cualquier punto de la aplicación. Permite la posibilidad de poner puntos de ruptura, “break points”, que permiten una ejecución paso a paso, ejecución hasta un punto determinado y se puede observar como los datos van tomando valores a medida que se va ejecutando la aplicación. Asimismo, lleva incorporado generadores de señales para poder hacer un simulador.

7.2. Introducción al Lab-VIEW

Labview es un lenguaje y a la vez un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla.

Nacional Instrument es la empresa desarrolladora y propietaria de LabView, se fundó en 1976 en Austin, Texas y sus primeros productos eran dispositivos para el bus de instrumentación GPIB. En abril de 1983 comenzó el desarrollo del Labview que vería la luz en abril de 1986 con la versión 1.0 para Macintosh, y en 1990 se lanzó la versión 2. La versión para Windows no llegó hasta 1992.

<i>Fecha</i>	Hito
Abril 1983	Comienza el desarrollo de LabView
Octubre 1986	LabView 1.0 para Mac
Enero 1990	LabView 2.0
Septiembre 1992	Labview para Windows
Octubre 1992	Labview para Sun
Octubre 1993	LabView 3.0 multilataforma
Abril 1994	LabView para Power Machkintos
Octubre 1995	LabView para Windows 95
Mayo 1997	LabView 4.0
Marzo 1998	LabView 5.0
Febrero 1999	LabView 5.1 LV para Linux y LV Rel-time
Agosto 2000	LabView 6.1
Mayo 2003	LabView 7 Express, LabView PDA y FPGA
Mayo 2004	LabView 7.1
Mayo 2005	LabView DSP
Junio 2005	LabView Embedded
Octubre 2005	LabView 8
Agosto 2006	LabView 8.2
Agosto 2008	LabView8.6
Enero 2009	Labview 9.0

Tabla V: Evolución de Labview

LabVIEW es un lenguaje completamente gráfico, y el resultado es que es totalmente parecido a un instrumento; por ello a todos los módulos creados con LabVIEW se les llama VI (Instrumento Virtual).

Tiene la característica de descomposición modular ya que cualquier VI que se ha diseñado puede convertirse fácilmente en un módulo que puede ser usado como una sub-unidad dentro de otro VI. Esta peculiaridad podría compararse con la característica de procedimiento en los lenguajes de programación estructurada.

Se trata de un sistema abierto, en cuanto a que cualquier fabricante de tarjetas de adquisición de datos o instrumentos en general puede proporcionar el driver de su producto en forma de VI dentro del entorno de LabVIEW. También es posible programar módulos para LabVIEW en lenguajes como C y C++; estos módulos son conocidos como sub-VIs y no difieren de los VI's creados con LabVIEW salvo por el interfaz del lenguaje en el que han sido programados. Además estos sub-VIs son muy útiles por ejemplo, en el campo de cálculos numéricos complejos que no se encuentran incluidos en las librerías de LabVIEW.

En la siguiente figura se muestra su pantalla de inicio.



Figura 73: Pantalla de inicio de LabVIEW

En cualquier VI existen dos conceptos bien diferenciados: el Front Panel (Panel Frontal) y el Block Diagram (Diagrama de Bloques):

- El Panel Frontal es la cara que el usuario del sistema está viendo cuando se está monitorizando o controlando el sistema, es decir, el interfaz del usuario. Además, puede ser totalmente parecido al instrumento del cual se están recogiendo datos; de esta manera, el usuario sabe de manera precisa cual es el estado actual de dicho instrumento y los valores de las señales que están midiendo. Este está formado por controladores e indicadores y existe una gran variedad de ellos, incluso se pueden diseñar controles e indicadores personalizados. Los primeros sirven para introducir parámetros al VI, mientras que los

indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.

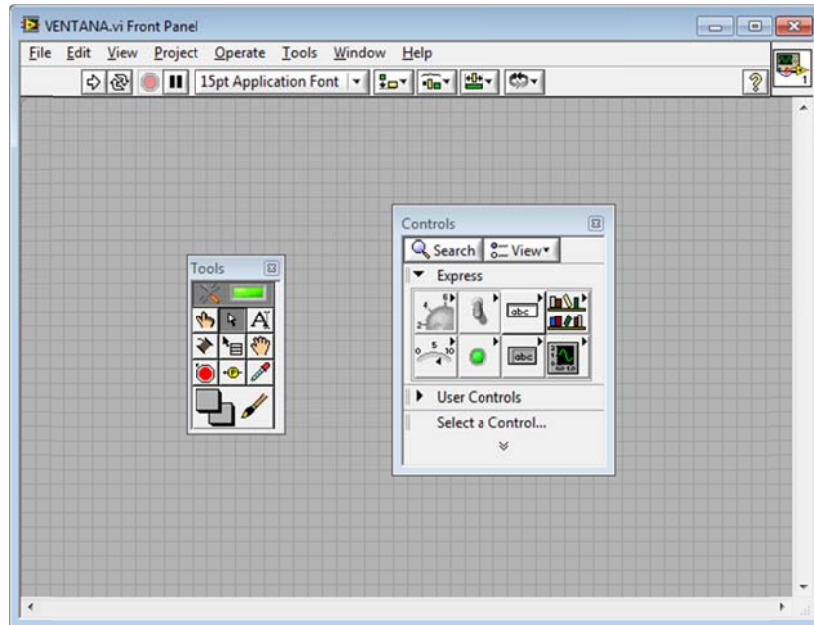


Figura 74: Panel frontal LabVIEW

- El Diagrama de Bloques del VI constituye el código fuente del VI, sería la cara oculta del Panel Frontal, una cara que el usuario del sistema no puede ver. En ella, es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesamiento de las entradas y salidas que se crearon en el panel frontal. El diagrama de bloques incluye funciones y estructuras integradas en las librerías que incorpora LabVIEW. En el lenguaje G las funciones y las estructuras son nodos elementales. Son análogas a los operadores o librerías de funciones de los lenguajes convencionales. Los controles e indicadores que se colocaron previamente en el Panel Frontal, se materializan en el diagrama de bloques mediante los terminales.

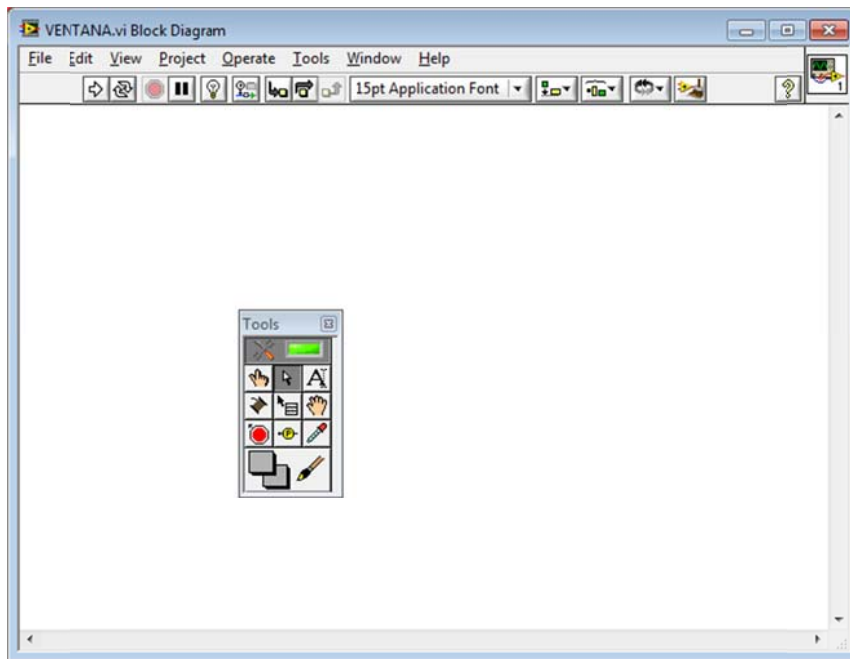


Figura 75 Diagrama de bloques

Todos los módulos están interconectados, mediante líneas de conexión, por donde circulan los diferentes datos o valores del VI. De esta manera, se logra que el VI funcione como un conjunto de elementos, módulos y sub-módulos.

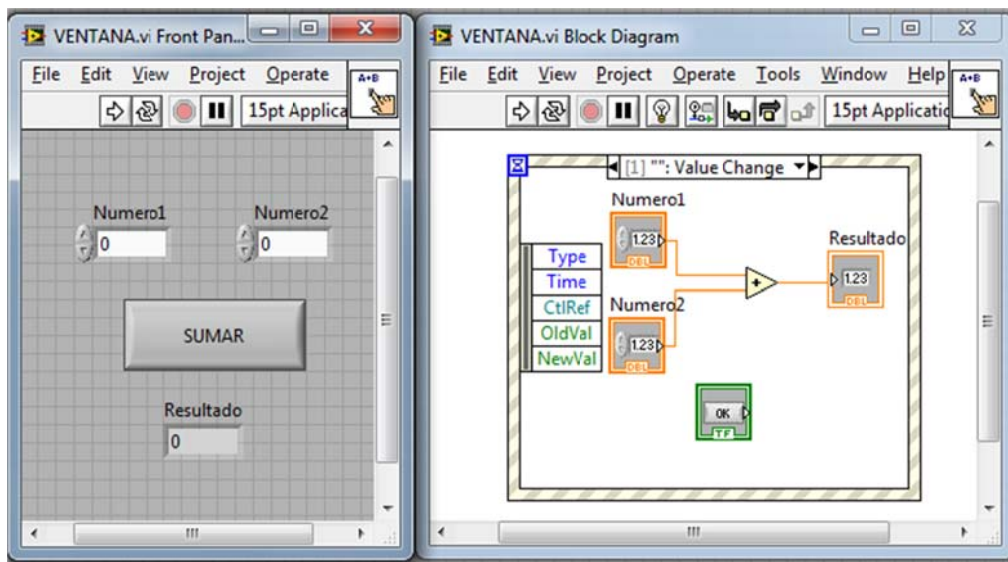


Figura 76: Panel frontal y diagrama de bloques

Los programas creados en LabVIEW reciben habitualmente el nombre de instrumentos virtuales o VIs, aunque no sean necesariamente aplicaciones de medición, test, etc. LabVIEW dispone de mecanismos para crear aplicaciones de prácticamente todo tipo.

La programación G (gráfica) de Labview consta de un panel frontal y un panel de código como se mencionó antes. En el panel frontal es donde se diseña la interfaz de usuario y se ubican los controles e indicadores. En el panel de código se encuentran las funciones. Cada control que se utiliza en la interfaz tiene una representación en el panel de código, igualmente los indicadores necesarios para entregar la información procesada al usuario tienen un icono que los identifica en el panel de código o de programación. Los controles pueden ser booleanos, numéricos, strings, un arreglo matricial de estos o una combinación de los anteriores; y los indicadores pueden ser como para el caso de controles pero pudiéndolos visualizar como tablas, gráficos en 2D o 3D, browser, entre otros.

Las funciones pueden ser VIs prediseñados y que pueden ser reutilizados en cualquier aplicación, estos bloques funcionales constan de entradas y salidas, igual que en un lenguaje de programación estándar las funciones procesan las entradas y entregan una o varias salidas, estos VI pueden también estar conformados de otros subVIs y así sucesivamente, de esta forma se pueden representar como un árbol genealógico donde un VI se relaciona o depende de varios SubVIs.

Labview tiene VIs de adquisición de datos e imágenes, de comunicaciones, de procesamiento digital de señales, de funciones matemáticas simples, hasta funciones que utilizan otros programas como Matlab o HiQ para resolver problemas, otras más complejas como "nodos de fórmula" que se utilizan para la resolución de ecuaciones editando directamente estas como en lenguajes de programación tradicionales y definiendo las entradas y las salidas. Labview también se puede utilizar para representaciones gráficas en tres dimensiones, en coordenadas polares y cartesianas, tiene disponibles herramientas para análisis de circuitos RF como la Carta de Smith, tiene aplicaciones en manejo de audio y se puede comunicar con la tarjeta de sonido del computador para trabajar conjuntamente. Entre sus muchas funciones especiales se encuentran las de procesamiento de imágenes, como capturar una imagen a través de una tarjeta de adquisición, analizarla y entregar respuestas que difícilmente otros sistemas realizarían.

En el desarrollo de aplicaciones para SADs (Sistemas de Adquisición de Datos) y sistemas con instrumentación destaca una herramienta que complementa al entorno LabVIEW™, MAX (*Measurement and Automation eXplorer*). De forma sencilla y rápida, con MAX se puede:

- testear instrumentos y dispositivos de adquisición conectados al terminal donde se ejecuta,
- gestionar conexiones y controladores de interfaz, los complementos y herramientas de desarrollo para LabVIEW que han sido instalados, los gestores de los instrumentos (*drivers*), etc.
- configurar el hardware de medida (por ejemplo, una tarjeta de adquisición de datos).

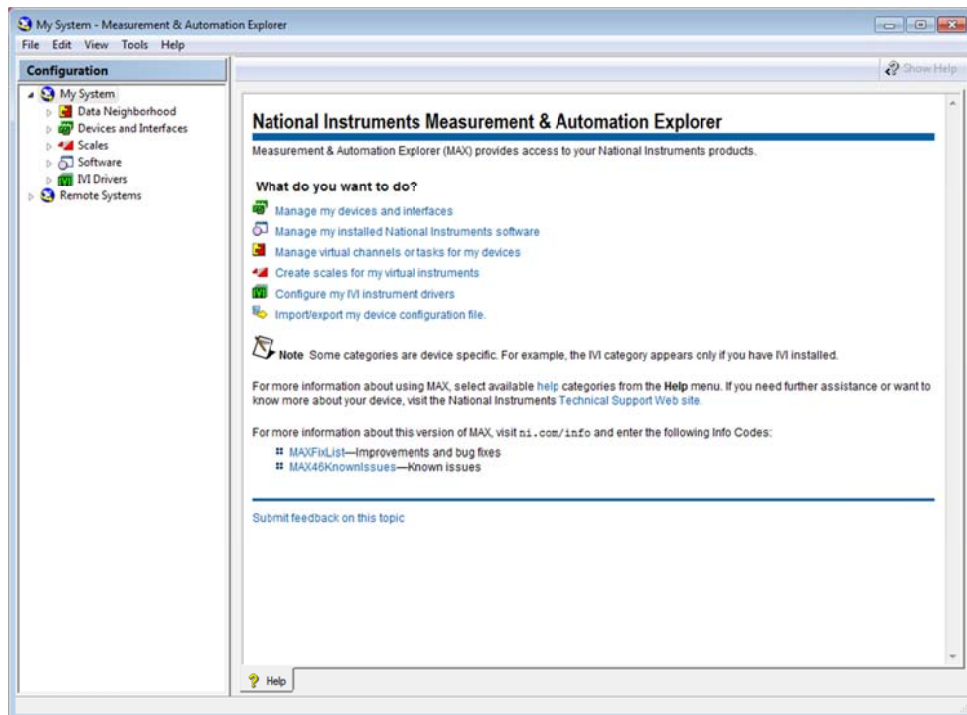


Figura 77: MAX (Measurement&Automation eXplorer).

7.3. ¿Cómo se programa en LabVIEW?

Para crear una aplicación básica en LabVIEW deberían seguirse al menos los siguientes pasos:

Lanzar el entorno de desarrollo (por ejemplo LabVIEW™ 2009). En consecuencia será posible observar que un VI consta de un panel frontal, un diagrama de bloques y su icono-conector.

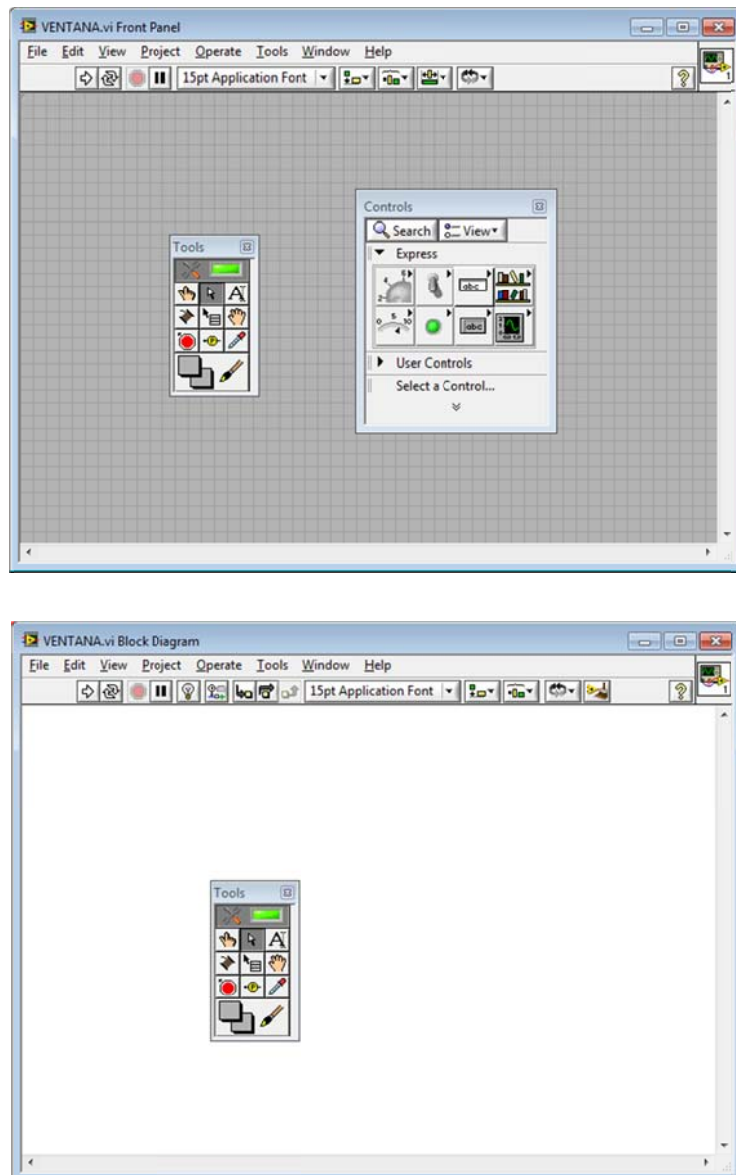


Figura 78: Aspecto inicial de un VI.

En el panel frontal y mediante controles e indicadores, elaborar la interfaz con el usuario. En los controles se introduce información; como pueden ser los parámetros para un experimento o la selección de una función a ejecutar. En los indicadores se visualiza otro tipo de información, como pueden ser los resultados de un experimento o los valores que va tomando una variable. Nótese que respecto a otras alternativas, LabVIEW proporciona un método muy rápido para construir interfaces gráficas de usuario (GUIs).

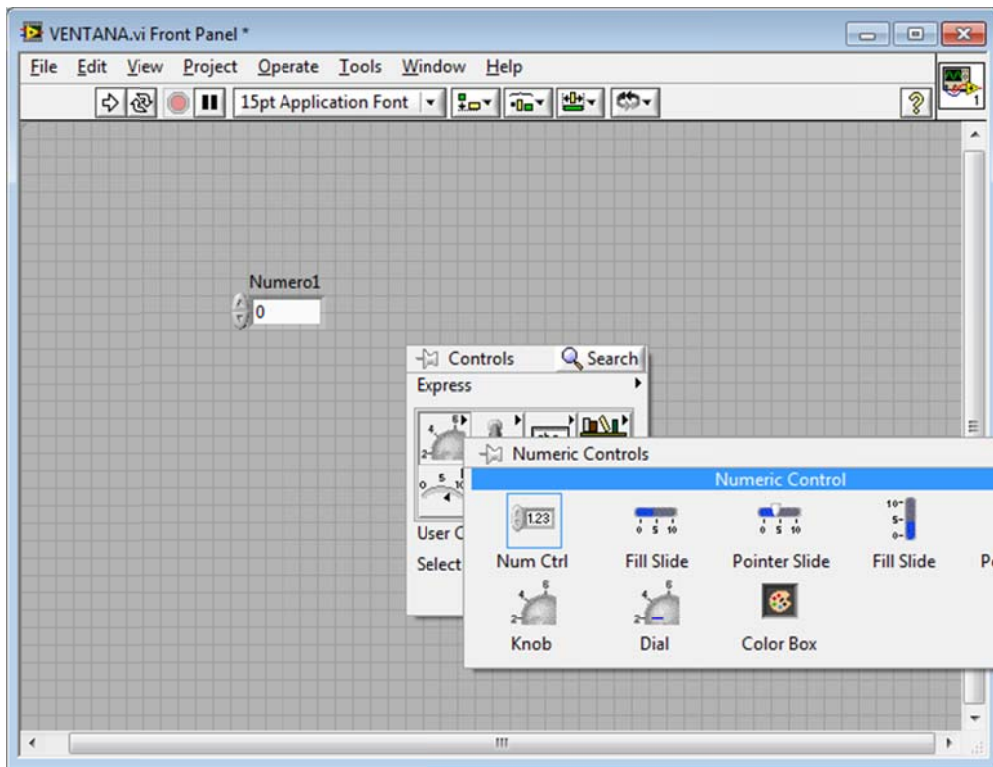


Figura 79: Elaboración de la interfaz de usuario (panel frontal).

En el diagrama de bloques, construir el código fuente. Para ello hay que insertar nodos y unirlos mediante cables. Los nodos equivalen a los datos y a las subrutinas del programa en un lenguaje basado en texto.

Por ello, los nodos de LabVIEW se encuentran en el código y son controles e indicadores, constantes, estructuras de control y VIs insertados en el código. Cada uno tiene unos terminales por donde recibe y da información, es decir, los terminales son sus entradas y salidas y además cada terminal pertenece a un tipo de dato. Finalmente, los nodos se conectan tendiendo cables entre terminales; un cable sale desde el nodo “fuente” de la información y termina en uno o varios nodos “sumidero”.

Un concepto muy importante es el de flujo de datos. En LabVIEW la ejecución del código es dependiente del flujo de datos, que en otras palabras significa que un nodo se ejecuta cuando hay datos disponibles en todos sus terminales de entrada, sino no. De la misma manera un nodo suministra los datos a todos sus terminales de salida cuando finaliza su ejecución. El flujo de datos es la forma más eficiente de transferir datos en LabVIEW, no obstante no es el único método y existen mecanismos para no respetarlo (se debe hacer un uso cuidadoso de los mismos).

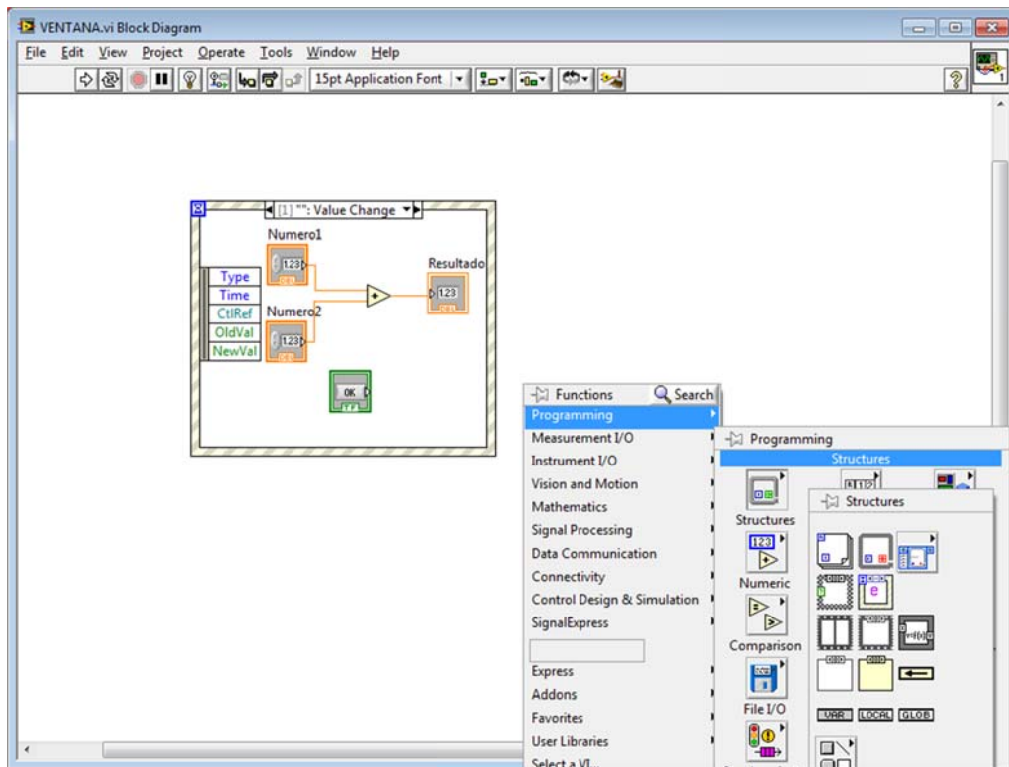


Figura 80: Elaboración del código fuente (diagrama de bloques).

Lo primero que se ha de hacer es depurar el programa y comentarlo.

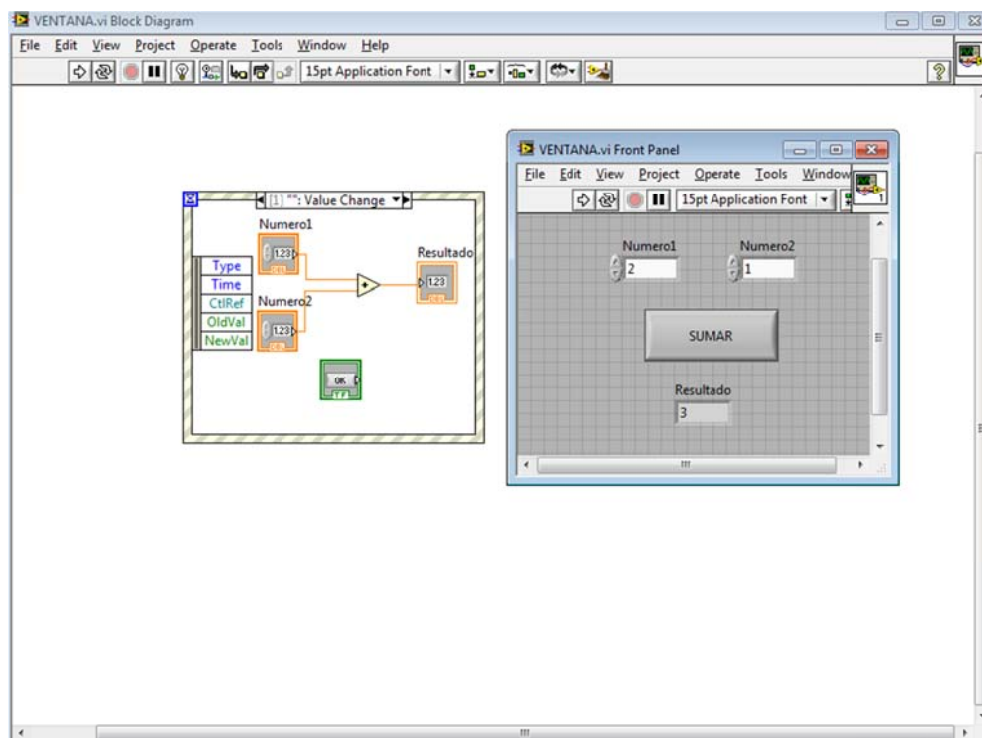


Figura 81: VI de ejemplo donde se ve: el panel frontal, el diagrama de bloques, el conector, el icono y comentarios.

Una vez finalizado el programa, hay que diseñar el icono del programa y establecer sus terminales, en nuestro caso al tratarse de una suma, tendremos 2 terminales de entrada y uno de salida.

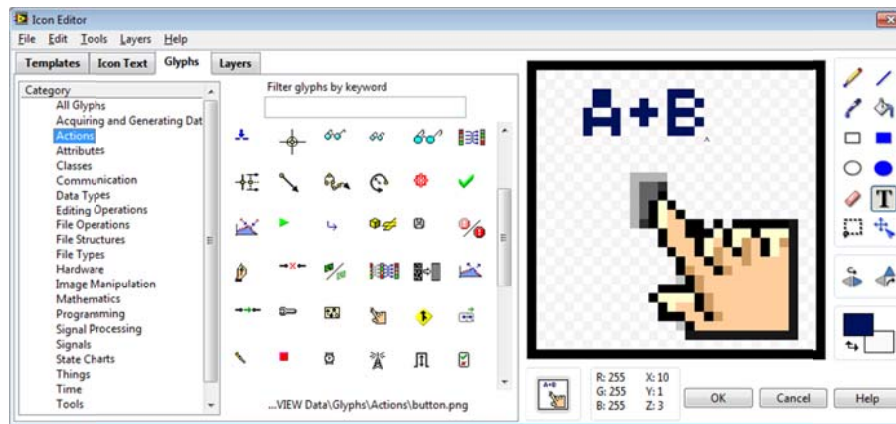


Figura 82 : Diseño del icono.

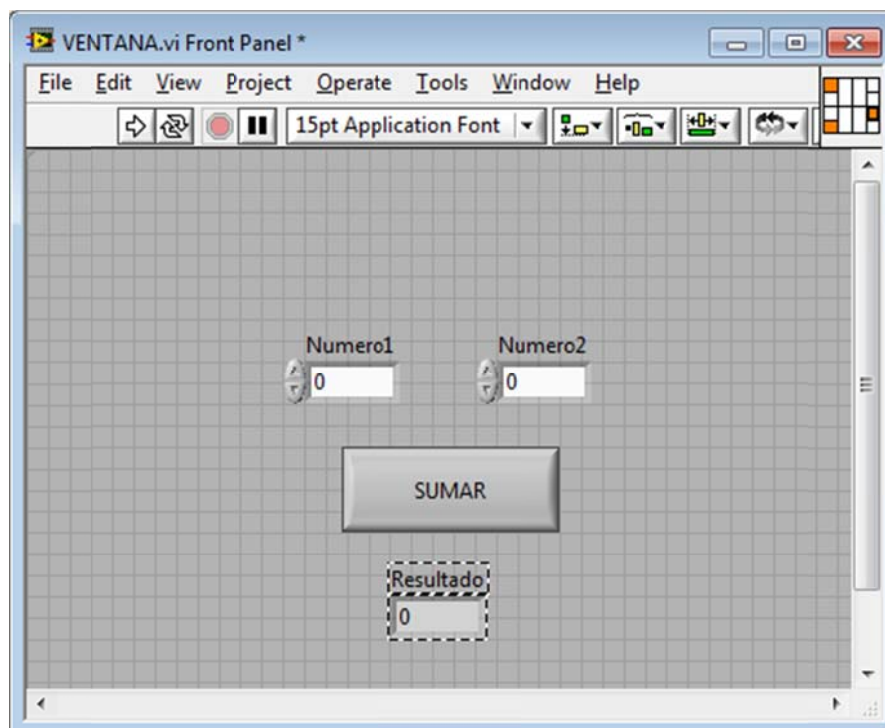


Figura 83 : Conexión de terminales.

7.4. VIs y estructuras de LabVIEW usadas

A pesar de que comprender un programa bien escrito en LabVIEW es sencillo, en comparación con otros lenguajes de programación, también requiere de un proceso de aprendizaje. A continuación se exponen solo aquellos elementos que es necesario mencionar, ya sea por útiles o por su habitual presencia en el proyecto.

7.4.1. Estructuras de control

Sirven para controlar la forma en que han de ejecutarse las instrucciones de un algoritmo. Las más típicas son las mencionadas a continuación, donde se puede apreciar que casi siempre llevan asociadas una condición que establece qué subcódigo seleccionar o cuántas veces iterar el subcódigo contenido (depende del tipo de estructura). En algunos textos se distingue entre estructuras de control y selección, aquí no, ya que al final ambos tipos sirven para modificar el flujo de instrucciones a ejecutar. Debe quedar claro que el flujo de datos también controla cómo han de ejecutarse las instrucciones, pero no es una estructura de control propiamente dicha.

1.1.1. Estructuras de control usadas

- Bucle *while*: (mientras que)



Figura 84 : Bucle While.

Ejecuta el código que contiene tantas veces mientras no se cumpla una condición de parada. La condición de parada debe establecerse empleando el símbolo que aparece abajo a la derecha en la representación.

El número de iteraciones no está prefijado, por lo que puede ser variable.

- Bucle *for*: (tantas veces)

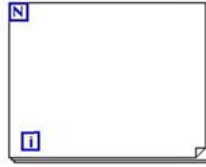


Figura 85 : Bucle for.

Ejecuta el código que contiene tantas como se indique. La cantidad de veces se indica empleando el símbolo que aparece arriba a la izquierda ("N").

El número de iteraciones está prefijado; pese a ello, también existe la posibilidad de establecer una condición de parada como en el bucle *while* (no es lo habitual, pero en ciertos momentos puede ser útil, como lo es ante la aparición de un error dentro del bucle).

- *Case*: (según el caso)

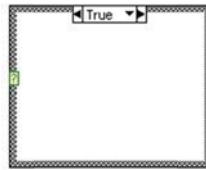


Figura 86 : Case.

Almacena distintos códigos en cada caso estipulado. Selecciona y ejecuta el caso que se le indique en su terminal de selección (símbolo "?", centrado a la izquierda). Requiere que se señale un caso por defecto, por si el caso indicado no coincidiera con ninguno de los estipulados.

Es una estructura equivalente a las estructuras *switch* e *if* de otros lenguajes de programación.

- *Event*: (según el evento)

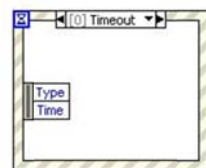


Figura 87 : Event

En el ámbito de este proyecto, los eventos son acciones producidas por el usuario o por el código sobre elementos del panel frontal.

Esta estructura espera hasta que se produce un evento de entre los estipulados, momento en que selecciona el código de ese evento (en esto es similar a la estructura *case*).

Según si se emplea o no el símbolo de arriba a la izquierda (“reloj de arena”), la espera tiene una duración o es eterna, respectivamente. Si se establece una duración y no se produce ningún evento, la estructura ejecuta el código que se le indique en el evento “*Timeout*” (tiempo de espera agotado).

1.1.2. Colas

Son un mecanismo para almacenar información momentáneamente y transferirla después, por lo que además de para transportar información pueden servir para sincronizar partes del código (nótese que esta acción quebranta el flujo de datos habitual).

Una cola maneja siempre el mismo tipo de dato (tipo de información). En LabVIEW concretamente, una cola es una fila de estos datos en la que los mismos se van añadiendo y retirando ordenadamente por cualquiera de los extremos (ver Figura 88, donde se compara con una fila de personas y un almacén de objetos). En consecuencia, son capaces de implementar estructuras tipo FIFO (*First-In/First-Out*) y tipo LIFO (*Last-In/First-Out*). En modo FIFO el dato añadido con mayor anterioridad es el primero en retirarse, por el contrario en modo LIFO el dato insertado más recientemente sería el primero en extraerse. Dado que el extremo por el que se desencolan (extraen) los elementos es el mismo para ambos tipos, la diferencia estriba en el extremo donde se encolan (añaden): el mismo extremo para encolar y desencolar en LIFO y extremos opuestos en FIFO. En general todas las colas creadas en este proyecto son de tipo FIFO (las que no lo son, es porque no se respeta completamente el mecanismo FIFO).

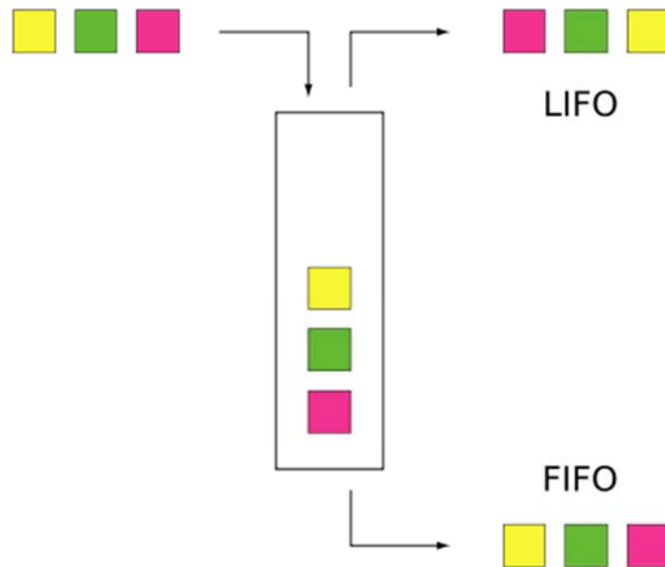


Figura 88. Estructuras FIFO y LIFO.

El procedimiento para emplear colas se basa en: crear previamente la cola, posteriormente encolar y desencolar datos, y finalmente liberar y destruir la cola por ser un mecanismo que usa memoria dinámica. Algunas funciones para operar con colas se muestran en la figura siguiente. Al crear la cola se obtiene una referencia a la misma, la cual es necesario pasar al VI que la utilice, para que sepa con qué cola debe trabajar.

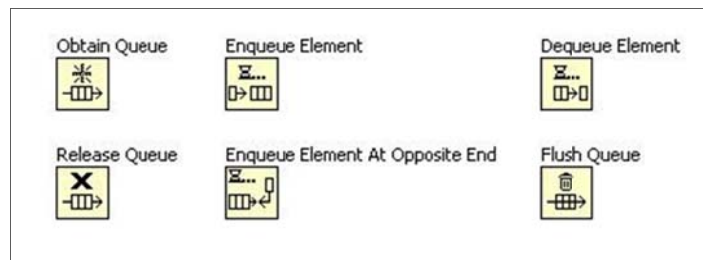


Figura 89. Funciones para operar con colas.

Conviene resaltar el comportamiento del VI “*Dequeue Element*”. Como bien indica su nombre se encarga de desencolar un elemento de la cola. Lo sobresaliente reside en que si no hay elementos en la cola, este VI espera indefinidamente a que exista uno (a no ser que se elimine la cola o se fije un tiempo de espera finito para el mismo). Esta característica lo hace muy interesante para ser usado en el consumidor de las arquitecturas productor-consumidor, concepto que se tratará más adelante.

1.1.3. Librería NI-DAQmx

Es un conjunto de funciones que constituye el *driver* del DAQ. Como puede apreciarse en algunos ejemplos de VIs, este conjunto de funciones ofrece herramientas para crear y especificar una tarea asociada a un DAQ, comenzar, parar o eliminar la tarea, escribir o leer datos en uno o varios canales especificados por la tarea, configurar otras opciones como la temporización, la señal de disparo, *streaming* a disco (los datos adquiridos se van almacenando directamente en un archivo), etc.

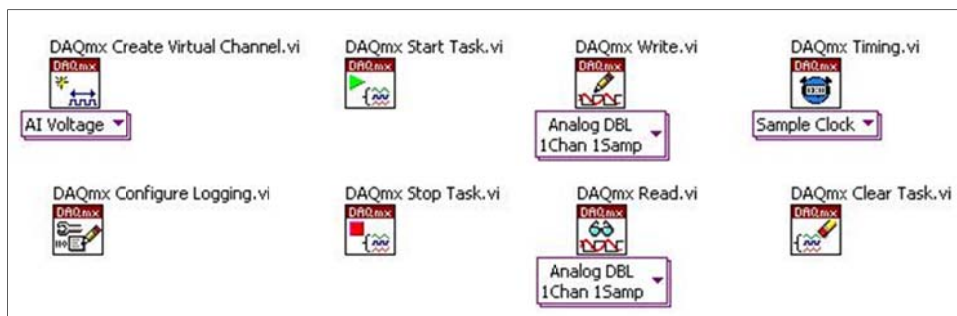


Figura 90. Funciones DAQmx.

El procedimiento habitual y más sencillo es crear una tarea primero (especificando entre otros, los canales a emplear), después configurarla (temporización, etc.), luego realizar el proceso de medida o generación de datos, y finalmente, cuando ya no se necesite, eliminar la tarea. Existen otras opciones dependiendo del fin que se persiga; a modo de ejemplo, las tareas podrían ser definidas directamente en MAX con sus características de escalado, canales empleados, etc.

Por último, por la misma razón que se mencionó anteriormente para los VIs para colas, y como también sucede con los VIs para ficheros o los pertenecientes a la librería VISA (presentados a continuación); los VIs de la librería NI-DAQmx necesitan que se les proporcione una referencia al elemento sobre el que tienen que operar. El elemento será una tarea en el caso de NI-DAQmx.

1.1.4. Librería VISA

Es un conjunto de funciones que componen el *driver* para la comunicación con equipos de instrumentación. Como puede apreciarse en algunos ejemplos de VIs, este conjunto de funciones ofrece herramientas para abrir y cerrar una sesión de comunicación con un instrumento determinado, transmitir o recibir datos a través del soporte de comunicación (por ejemplo, un bus GPIB), configurar otras opciones,

algunas propias del soporte de comunicación (por mencionar un caso, el establecimiento de la velocidad de transmisión en una comunicación RS-232), etc.

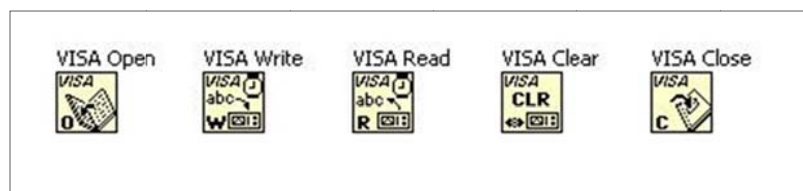


Figura 91. Funciones VISA.

El procedimiento básico de operación es similar al indicado para los VIs de NI-DAQmx y de colas: abrir/crear, usar y cerrar/destruir. En el caso de las funciones VISA el elemento referenciado es una sesión de comunicación, en la que van implícitas las peculiaridades de la comunicación.

1.1.5. Archivos: binarios, texto, Datalog, y TDMS.

Los archivos o ficheros se utilizan para almacenar y organizar información de todo tipo. Cada clase de archivo tiene su propia estructura y por lo cual, determinadas características; razón por la que conviene elegir el tipo de archivo adecuado según las particularidades de la información y el uso que se la vaya a dar. A continuación se hace un resumen con fines prácticos de los tipos de archivo empleados por la aplicación construida.

Los archivos binarios son el tipo de fichero básico por excelencia; en ellos es posible almacenar cualquier tipo de información e incluso tipos distintos. Pueden llevar una cabecera con metadatos o no, en este último caso se dice que son “archivos binarios planos”. Los metadatos son datos que describen los datos guardados, por ejemplo indicarán si es una imagen, un sonido, etc. u otros atributos (fecha de creación, usuario que lo creó, etc.).

Los ficheros binarios tienen como ventajas: ocupar mínimos de espacio en disco, no tener la necesidad de formatear los datos cuando se leen o escriben, acceder rápidamente a los datos, posibilidad de añadir y obtener información aleatoriamente, y posibilidad de emplear números con gran precisión. Las principales desventajas es que sólo son legibles por las máquinas (es decir, para los humanos no son directamente legibles) y que para poderlos utilizar es necesario conocer la estructura del fichero y los tipos de datos que guardan.

Los ficheros de texto son específicos para almacenar texto. El archivo de texto básico es el llamado “archivo de texto plano”, que contiene simplemente una secuencia de caracteres ASCII. La ventaja principal es que los datos son legibles directamente por las personas, y por ello y por su sencillez es el tipo más intercambiable, tanto entre personas como entre aplicaciones. Las desventajas frente a otros, como los ficheros binarios, son que ocupan mayor espacio en disco, la velocidad de transferencia es menor (sobre todo si los datos se tienen que formatear previamente), la precisión se reduce y es difícil acceder a los datos aleatoriamente porque el espacio en memoria que requieren no es fijo.

Por lo tanto, se recomienda emplear archivos binarios cuando se guarde información del mismo tipo o cuando la estructura del archivo es muy conocida; y en especial se recomiendan si la cantidad de datos es grande (un ejemplo sería al tomar medidas de un voltaje cada milisegundo a lo largo de un día). Los ficheros de texto son aconsejables sólo cuando es necesario compartir la información (sobre todo si el usuario destino no tiene acceso a la aplicación que creó el fichero, aunque existen otros formatos) o en ocasiones especiales, como cuando se quiere almacenar la secuencia de instrucciones a enviar a un instrumento programable (recuérdese que se transmiten mensajes en formato ASCII). En la siguiente hay VIs para el manejo esencial de estos ficheros.

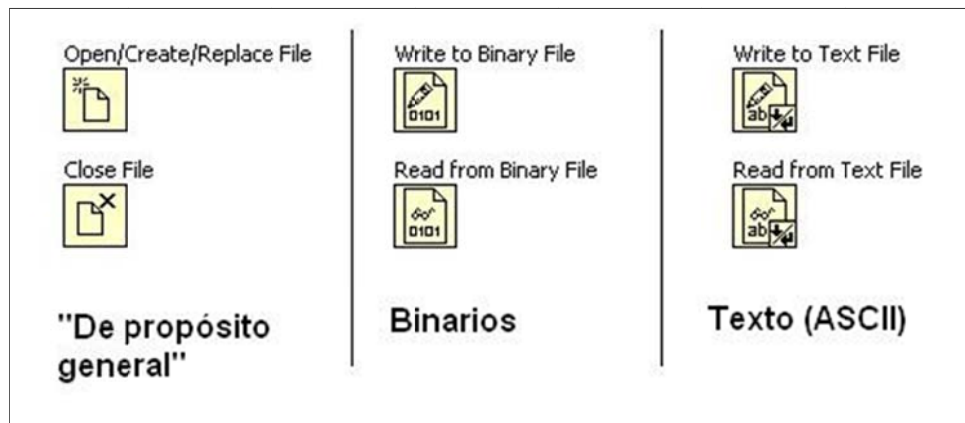


Figura 92. Funciones para tratar archivos binarios y de texto.

Los archivos “Datalog” son aproximadamente un caso especial de fichero binario. Se emplean cuando es necesario almacenar información en formato binario y la información posee una estructura compleja (por ejemplo, un largo *array* de un tipo *clúster* con diversos tipos de dato), ya que lo que hace es almacenar y organizar tal información en una secuencia de registros idénticos.

Los pros que presentan se pueden considerar iguales a los de los ficheros binarios, con la salvedad de que en los Datalog el acceso aleatorio a los datos y su manipulación es mucho más sencilla y rápida, ya que cada registro está determinado por un número. La desventaja aparece si se desea modificar la aplicación y una clase de ficheros Datalog se vea afectada por el cambio, porque la aplicación no conseguirá recuperar correctamente la información de los ficheros antiguamente creados (por ejemplo, esta situación se da cuando se incluye un nuevo dato en la estructura de datos sobre la que trabaja el Datalog). La siguiente muestra VIs para archivos Datalog.



Figura 93. Funciones Datalog.

El formato de fichero TDMS (*Technical Data Management Streaming*) es otro tipo de fichero binario con la particularidad de que está optimizado para aplicaciones de test y medida. Se ha diseñado de manera que sea fácilmente intercambiable, estructurado internamente pero escalable, con un acceso sencillo a la información y con capacidad para acceder muy rápidamente a los datos del fichero (posibilidad de hacer *streaming* con los datos). En la siguiente figura se tienen VIs para manipular archivos TDMS.

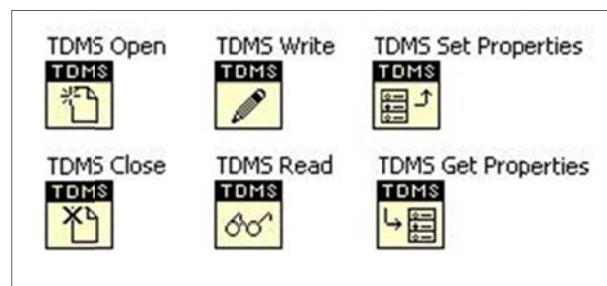


Figura 94. Funciones TDMS.

Es de particular interés la estructura de un archivo TDMS cuando se almacena información de test y medidas. Por un lado, el archivo está organizado jerárquicamente en tres niveles: fichero, grupos y canales. El nivel de fichero puede contener un número ilimitado de grupos y a su vez un grupo puede contener un número ilimitado

de canales. Por otro lado, cada elemento de nivel (archivo, grupo o canal) puede almacenar un número ilimitado de propiedades o atributos.

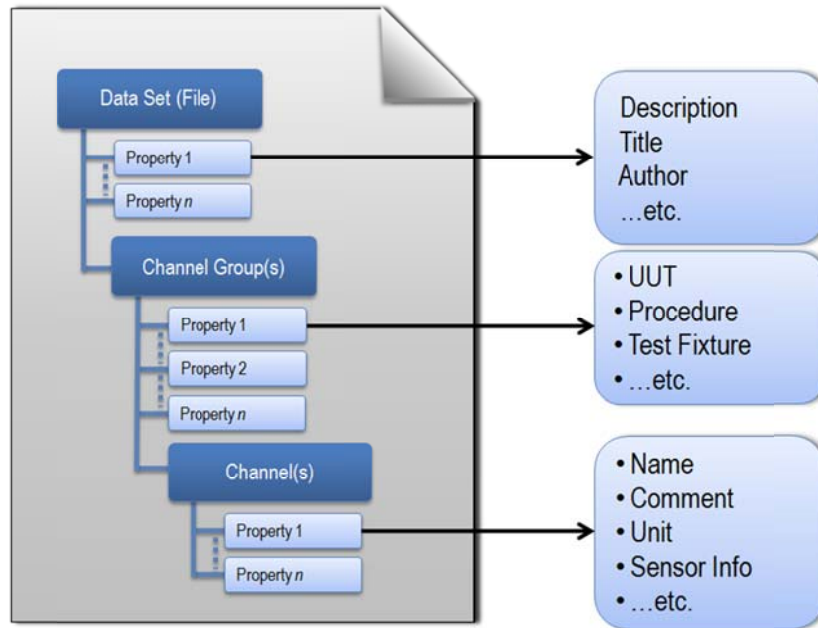


Figura 95: Estructura genérica de un archivo TDMS.

1.1.6. Vis propios

Para la construcción de la aplicación se han construido diversos VIs propios, que no existían previamente. Cada uno de ellos cumple una función bien determinada en la aplicación, por lo que han servido para modularizar el programa y optimizar la reusabilidad de los códigos implementados.

Estos Vis, se explicarán detenidamente en el apartado de diseño software, de tal forma que un futuro usuario no tenga ningún problema en identificar todos los componentes del programa de control.

1.1.7. Máquinas de estados

En general, las máquinas de estados son uno de los patrones de diseño comúnmente más utilizados. Gozan de una flexibilidad y escalabilidad sorprendente, por lo que su “abuso” se recomienda en aplicaciones de mediana y alta complejidad.

En una máquina de estados el objetivo a conseguir se fragmenta en un conjunto de estados que realizan cometidos más sencillos. El avance del programa se consigue mediante transiciones de estado, que pueden ser directas entre dos estados o basadas en un proceso de decisión cuando el paso al siguiente estado, de entre

varios posibles, atiende a las condiciones existentes. Es muy habitual representar las máquinas de estados mediante diagramas de bolas, tal como se hace en la Figura 96. Diagrama de bolas de una máquina de estados de ejemplo. que será utilizada como ejemplo.

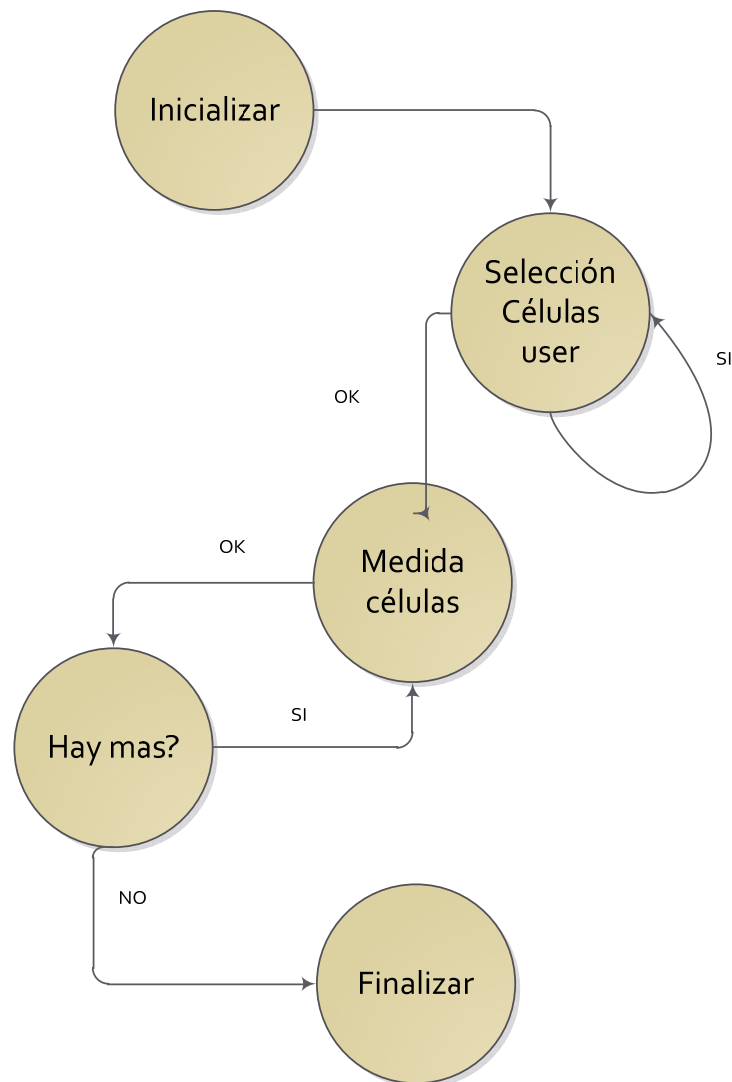


Figura 96. Diagrama de bolas de una máquina de estados de ejemplo.

Como puede apreciarse en el diagrama de bolas, cada bola (círculo) simboliza un estado y cada flecha significa una transición. Las acciones detalladas de cada estado se concretan en un lugar aparte, en las bolas se escribe una frase representativa simplemente. Luego, si la transición es condicionada tienen que partir varias flechas del estado origen a los estados destino y en la flecha suele indicarse la condición a cumplir, aunque a veces se indican aparte mediante tablas cuando el número de condiciones es tan grande que imposibilita el dibujo. Si la transición es directa, sólo sale una flecha del estado origen al estado destino.

El diagrama de bolas principal puede descomponerse recursivamente en un conjunto de diagramas cuando existen demasiados estados, simplemente es necesario especificar una bola que represente al sub-diagrama en el diagrama principal y en otro lugar dibujar el sub-diagrama. También, en cualquier momento se puede escalar la aplicación definiendo más estados e intercalándolos entre los existentes, o bien, lo contrario que es eliminar estados.

En LabVIEW hay diversas maneras de montar máquinas de estados, estando la mayoría compuesta por un bucle *while* y un *case*. La diferencia radica normalmente en la forma en la que se transita entre estados, por ejemplo en la máquina de estados.

Puede hacerse de una forma clásica o estándar (Figura 97. Patrón de una máquina de estados estándar.): se emplean tipos definidos *enum* y *shift registers*.

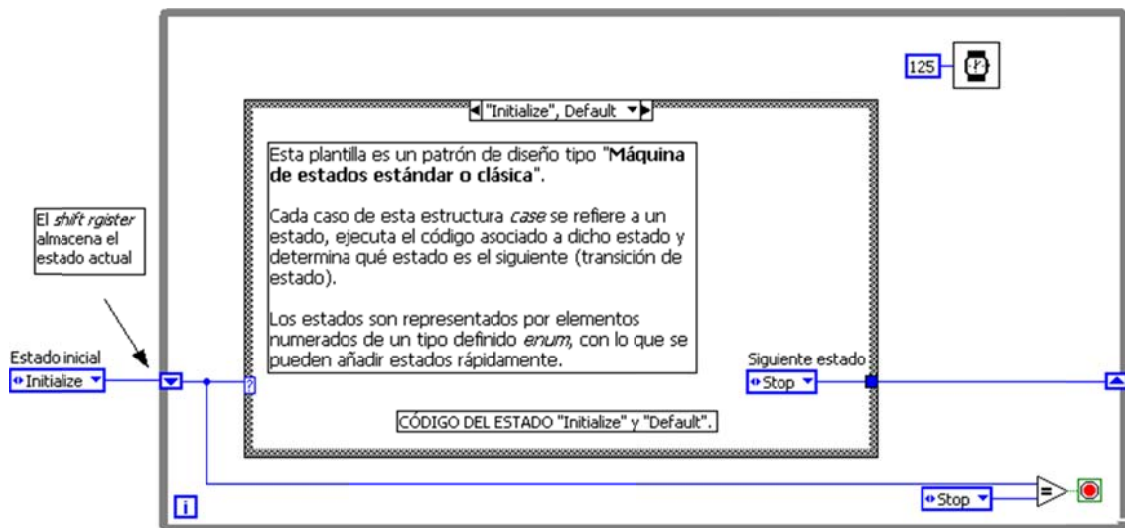


Figura 97. Patrón de una máquina de estados estándar.

Puede hacerse por colas (Figura 98. Patrón de una máquina de estados mediante colas.), o en su lugar, *arrays*: en el caso de colas se "auto" encolan y desencolan estados en la misma máquina creando una secuencia de estados. De forma similar, en el caso de *arrays* se insertan o eliminan estados con el mismo objetivo.

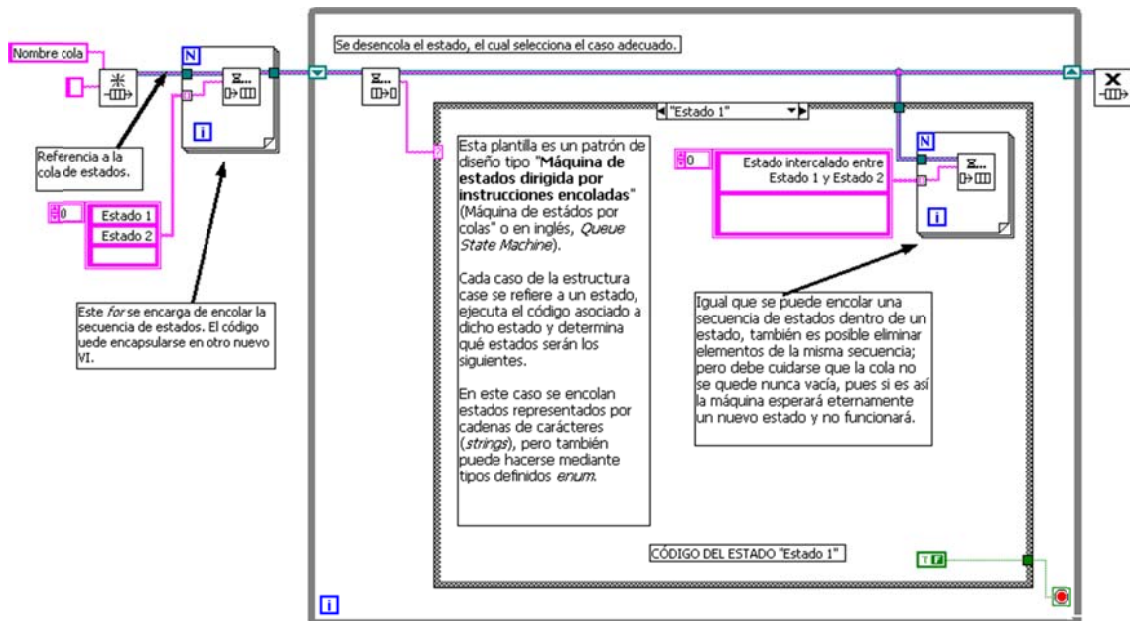


Figura 98. Patrón de una máquina de estados mediante colas.

Y por último, puede ser conducida por eventos atendiendo a los eventos que se producen se cambia de estado.

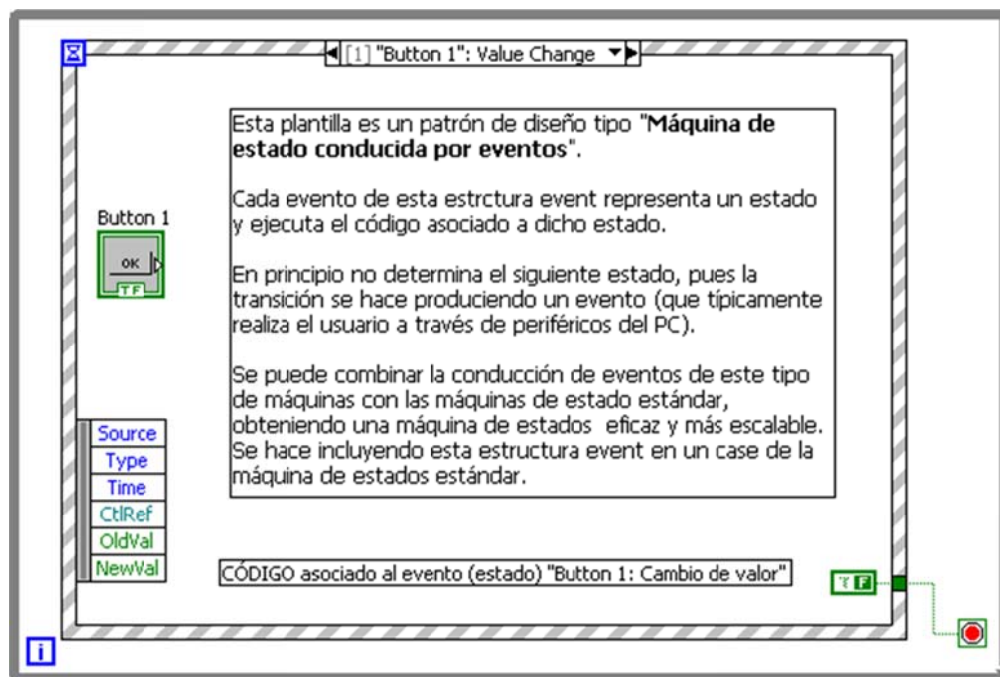


Figura 99. Patrón de una máquina de estados conducida por eventos.

1.1.8. Control de una máquina de estados

Además de los controles disponibles en el menú control, se pueden crear otros nuevos mediante la modificación de uno existente, para ello primero hay que colocar

en el Panel Frontal un control del tipo de datos deseado y después elegir en el menú Edit >Customize Control.

Cuando se coloca un control personalizado en un Panel Frontal no existe ningún vínculo entre el fichero donde el control está definido (*.ctl) y sus instancias empleadas en cada VI donde es usado, se trata de copias independientes. Si una instancia se modifica no afecta al resto. Las definiciones de tipos y definiciones de tipos estrictos (Type Def. y Strict Type Def.) se usan para enlazar todas las instancias de un control o indicador a un mismo fichero *.ctl. De esta forma al modificar el fichero se modifican todas las instancias.

Las Type Definitions definen el tipo de datos de un control. Cuando el tipo de datos cambia, todos los controles asociados a él también cambian. En el menú contextual de una instancia se puede seleccionar Auto-Update from Type Def. para desactivar la actualización automática.

Un Strict Type Definition hace que todas las instancias sean iguales, no sólo el tipo de datos, también en características como el rango, tamaño, color, etc. De esta forma si se modifica la definición se actualizarían todas las instancias. Desde el menú contextual se puede eliminar el enlace entre la instancia y la definición.

1.1.9. Productor-consumidor

Un bloque (de código) productor es aquel que genera órdenes o datos para otro bloque, el cual sería el consumidor (de esas órdenes o esos datos). Una arquitectura productor-consumidor está formada por uno o varios bloques productores y uno o varios bloques consumidores, con la posibilidad de que un bloque sea a su vez consumidor de uno y productor de otro. La arquitectura productor-consumidor más sencilla es la que está formada por un productor y un consumidor.

En LabVIEW el productor y el consumidor se implementan normalmente con bucles del mismo tipo (casi siempre bucles *while*, ver Figura 100. Ejemplo patrón de un productor-consumidor) y la comunicación entre ellos se hace mediante colas. El código dentro de estos bucles no suele tener un formato común, pues depende de la forma en la que el productor “produce” para el consumidor y el tipo de producto (datos u órdenes) que “consume” el consumidor. Así, en el caso de que el consumidor tome órdenes dadas mediante un *string* o un *typedef enum* el consumidor incorporará un *case*, y en el caso de que el productor genere órdenes atendiendo a eventos del usuario incorporará una estructura *event*.

Para definir una arquitectura productor-consumidor en la etapa de diseño o en la de documentación, se puede recurrir a hacer tablas cuando el consumidor recoja órdenes. En la tabla se especifica la orden y la acción que debe realizar el consumidor cuando la despache (dado que las órdenes se encolan, éstas tienen un orden al ser desencoladas).

Se puede especificar simplemente el tipo de datos que recoge el consumidor, pues este tipo de datos siempre será el mismo. Un ejemplo de caso es cuando un productor mide adquiriendo conjuntos de muestras de una señal y el consumidor debe graficar en pantalla dichos datos que el productor genera.

Esta plantilla es un patrón de diseño tipo "Productor/Consumidor".

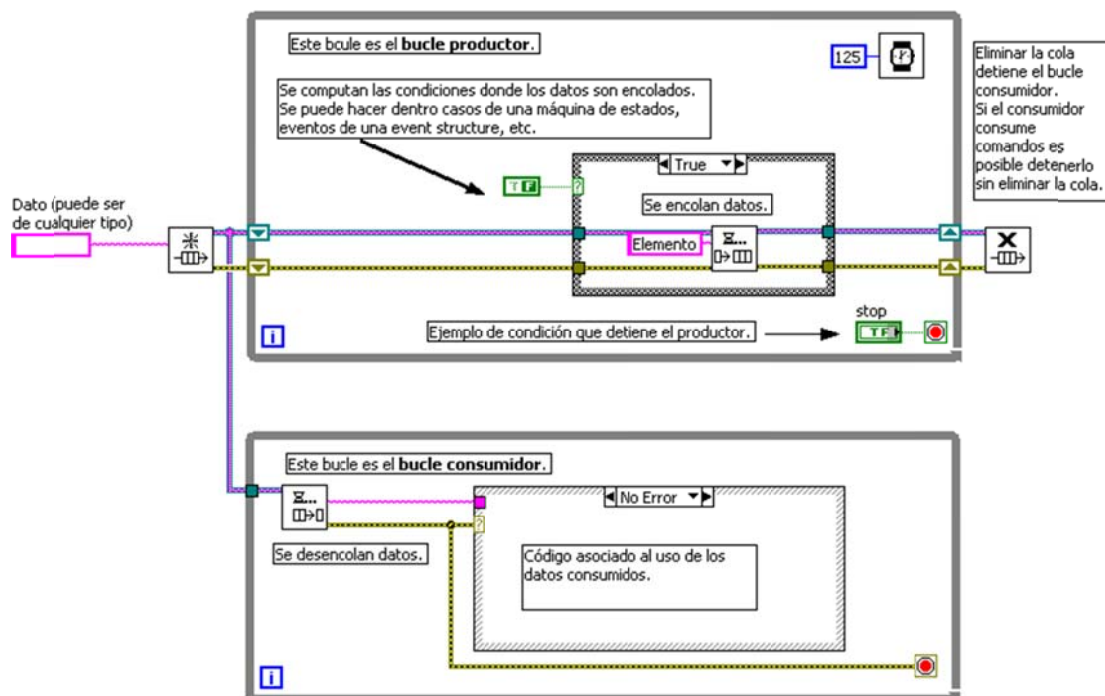


Figura 100. Ejemplo patrón de un productor-consumidor.

Es importante darse cuenta de las siguientes características:

- Un bloque que sólo hace de productor se ejecuta continuamente y puede generar información para el consumidor (datos u órdenes) en cualquier momento; sin embargo, cualquier consumidor sólo puede ejecutarse cuando hay información en su cola de espera.
- Las colas dan soporte al intercambio (comunicación) entre productor y consumidor, pero a su vez garantizan que la sincronización sea correcta entre ambos bloques de código.

- Ninguna información una vez producida, no es consumida; es decir, no se pierden acciones. Esto a veces se traduce en una ventaja respecto a la arquitectura maestro-esclavo, que es parecida pero no lo hace. De todos modos, si se manipulan apostando los elementos ya existentes en la cola es posible desobedecer esta característica con criterio.
- Cuando el consumidor concluye sus cometidos más lentamente que lo que tarda el productor en producir, la cola corre el peligro de saturarse (crece indefinidamente hasta alcanzar una cantidad límite de elementos). Ésta es una desventaja de la arquitectura productor-consumidor y hay que prestarla atención.

1.1.10. Combinaciones

A través de la combinación de patrones de diseño como los mencionados anteriormente, pueden fabricarse arquitecturas *software* más complejas pero a su vez fáciles de desglosar por el mecanismo inverso. Como bien se verá en el diseño, numerosas máquinas de estado y múltiples relaciones productor-consumidor formarán una arquitectura compleja final.

7.4.2. Buenas prácticas de programación con LabVIEW

La experiencia de los programadores ha resaltado la importancia de cumplir unas convenciones y procedimientos a la hora de programar. Estos acuerdos han dado lugar a los llamados estilos de programación correctos (a veces nombrados como estándares o convenios de código). Los estándares de código son con frecuencia dependientes del lenguaje de programación utilizado y LabVIEW, como lenguaje que es, posee los suyos (Blume, 2007).

Es importante secundar estas normas por los siguientes motivos:

- Alguien que no es autor del código debe utilizarlo.
- Se deja un proyecto por un tiempo y más tarde es necesario retomarlo.
- Un cliente rechaza la aplicación por no cumplir ciertas normas o porque no le gusta su aspecto.

- Se carece de tiempo para arreglar problemas que puedan surgir en un futuro.

Por eso las guías de estilo básicas para LabVIEW definen:

- Cómo abordar un proyecto: especificaciones, diseño, configuración del entorno, uso de las herramientas de desarrollo, organización, etc.
- El aspecto que ha de tener la interfaz de usuario: textos legibles, colores cómodos, funcionalidades, etc.
- Cómo construir el diagrama de bloques: separación de cables, visibilidad de los nodos, comentarios del código, etc.
- Cómo crear conectores e iconos adecuados: iconos significativos, conectores similares en una familia de VIs, evitar la saturación de conexiones en el conector, importancia de cada terminal (obligatoria, recomendada u opcional), etc.
- El uso de las estructuras de datos: estructuras eficaces y representativas, metodología para elaborarlas, empleo de tipos definidos (manipulables o estrictos) en ciertas ocasiones, etc.
- La relevancia de preocuparse por los errores: tratamiento, propagación, priorización y tipificación de los errores, estructura como dato de un error, etc.
- La practicidad del uso de patrones de diseño.
- La importancia de comentar y documentar correctamente todo lo necesario: diagrama de bloques, elementos del panel frontal, funciones del VI, versiones y revisiones del programa, etc.

Sin entrar en más detalles de la ingeniería del *software*, se subraya también la gran importancia de utilizar y combinar técnicas propias de diferentes paradigmas de programación y diseño algorítmico, la modularidad, la abstracción, la herencia, etc.



Figura 101: Clasificación general de los paradigmas de programación.

La elaboración de este proyecto se ha apoyado principalmente en la guía de estilo de Blume (2007) y los artículos publicados en la página web de National Instruments (Instrument). Blume y otros expertos definen que se persigue una aplicación fácil de usar y actualizar, escalable, eficiente, legible, simple, productiva, mantenible y robusta.

8

Arquitectura Software

8.1. Diagrama de estados.

Lo primero que haremos antes de ponernos a programar será definir nuestro diagrama de estados, en dicho diagrama se contemplan todas las acciones necesarios para llevar a cabo el procedimiento de medida y almacenamiento de datos.

8.1.1. Primera aproximación

El diseño consiste en 3 estados fundamentalmente, como se ha descrito en el apartado hardware, el primero de ellos consistirá en el estado de reposo, en el cual nuestro circuito no estará haciendo nada, las fuentes de corriente estarán apagadas, y la fuente de alimentación para tomar las medidas también. Después del estado de reposo pasaremos al estado de estrés, en que seleccionaremos nuestros relés en la posición estrés y encenderemos las fuentes de corriente, en dicho estado deberemos permanecer el tiempo seleccionado para estresar la célula entre medida y medida. Una vez estresada dejaremos reposar de nuevo la célula y pasaremos a tomar las medidas correspondientes para poder estimar en un futuro la degradación sufrida.

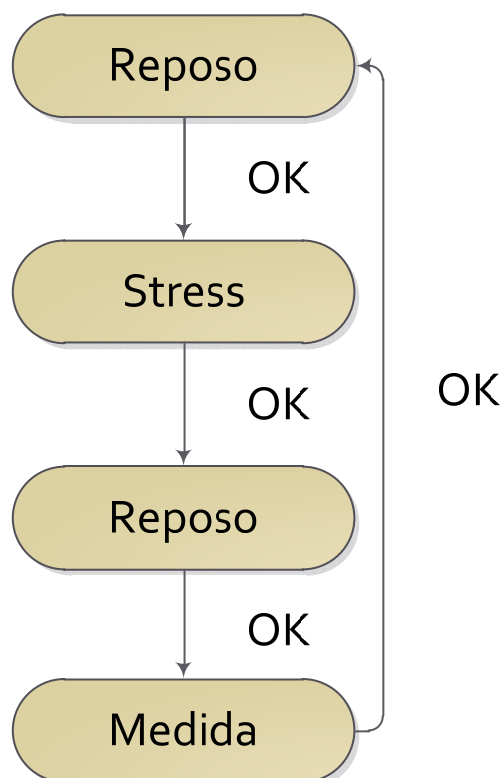


Figura 102: Diagrama de estados, primera aproximación.

8.1.2. Segunda aproximación

La arquitectura software del sistema estará basada en una máquina de estados, basada en la ejecución en paralelo de distintos gestores.

La máquina de estados, sigue el siguiente esquema:

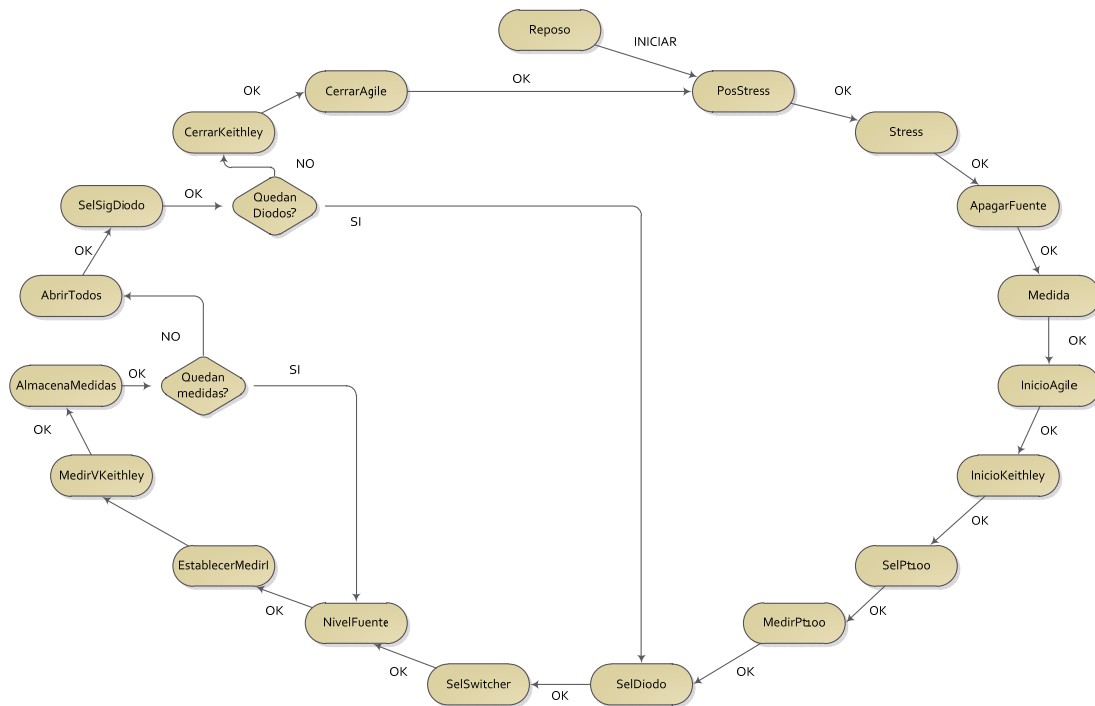


Figura 103 : Diagrama de estados, segunda aproximación.

Reposo: Este es el estado inicial en el que entra al arrancar el programa, aquí esperará hasta que el usuario pulse iniciar.

PosStress: Selecciona todos los relés en posición Stress.

Stress: Enciende la fuente corriente para comenzar a someter a stress a la célula, y espera en este estado hasta que transcurre el tiempo de estrés seleccionado por el usuario, además en este estado se guarda configuración del usuario por si en un futuro deseara usarla.

Apagar Fuente: Apaga la fuente de corriente para dejar de someter a estrés a la célula.

Medida: Selecciona todos los diodos en posición stress, y realiza una espera equivalente al tiempo de estabilización establecido por el usuario.

InicioAgile: Inicializa el Agile.

InicioKeithley: Inicializa el Keithley.

SelPt100: Selecciona el canal del switcher Keithley destinado a medir la temperatura dentro del horno para asegurar que el horno está funcionando correctamente.

MedirPt100: Mide la resistencia Pt100 y establece la temperatura a la que se encuentra el horno.

SelDiodo: Selecciona una célula a medir y la coloca en la posición medir.

SelSwitcher: Selecciona el canal del switcher asociado a la célula seleccionada.

NivelFuente: Establece la tensión seleccionada con la Fuente Agilent.

EstablecerMedirI: Mide y almacena el valor de la corriente con la fuente Agilent.

MedirVKeithley: Medimos la tensión en bornas de la célula con el switcher keithley.

Almacena Medidas: Almacena todas las medidas tomadas.

AbrirTodos: Abre todos los canales del switcher para asegurarse que ninguno está seleccionado.

SelSigDiodo: Selecciona la siguiente célula si la hubiera.

CerrarKeithley: Termina la comunicación con el switcher.

Cerrar Agile: Termina la comunicación con la fuente Agile.

8.2. Gestores

El diseño es una estructura de gestores conforme al siguiente esquema. Como bien se comprueba en la figura, estos gestores comparten datos mediante variables globales y se comunican mediante colas.

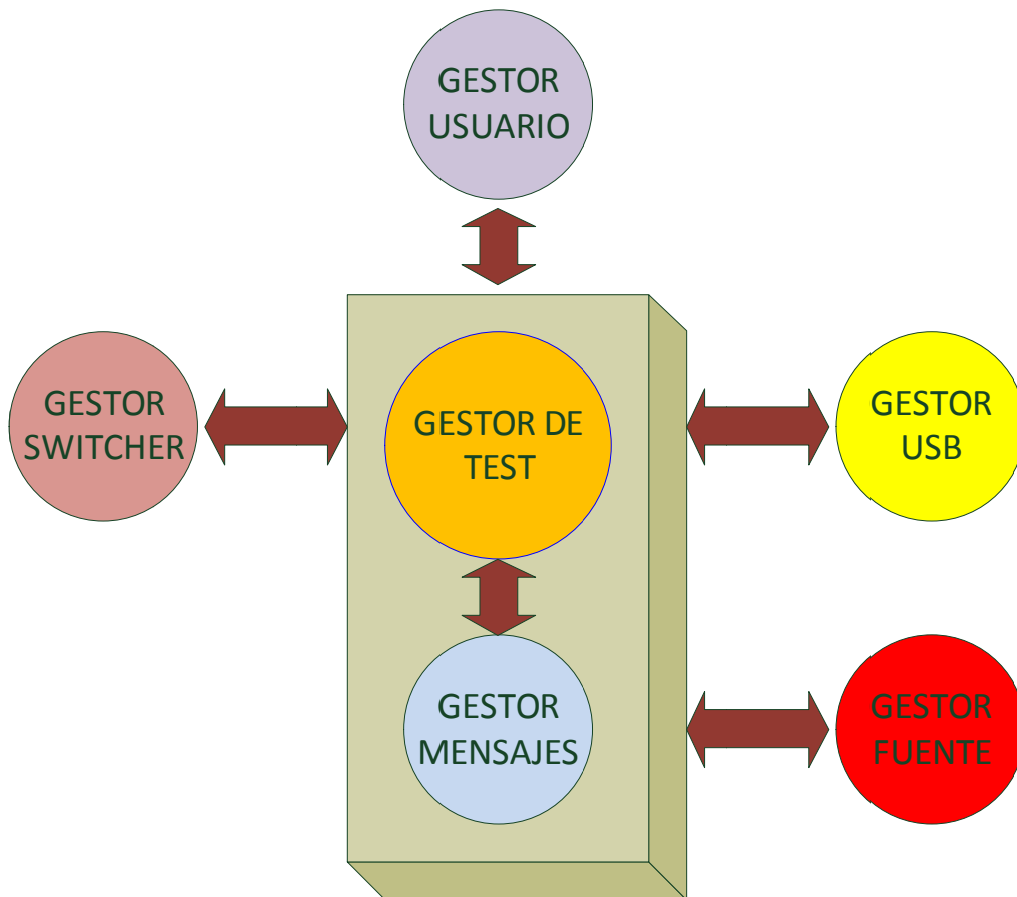


Figura 104 : Estructura Gestores.

8.2.1. Gestor de usuario

Gestiona la interfaz del módulo con el usuario. Debe abordar dos funciones fundamentales: atender a las acciones del usuario (eventos generados por pulsaciones de teclas, clics del ratón, etc.) y el módulo encargado de la presentación de datos en pantalla (menús, gráficos, controles, indicadores, cuadros de diálogo, etc.). Está integrado en el VI principal de la aplicación.



Figura 105 : Gestor usuario

8.2.2. Gestor de Test

Es nuestra máquina de estados, el gestor de test se encarga del control de todo el programa, enviando las órdenes al resto de gestores, indicándolos en cada momento la acción a realizar, este a su vez recibe la respuesta de los resultados de realizar las acciones. Es el único VI del sistema que posee capacidad de decisión.

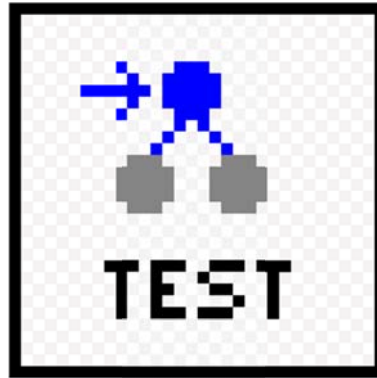


Figura 106 : Gestor test.

8.2.3. Gestor Mensajes

Por este gestor pasan todos los mensajes que circulan por el programa, es el encargado de comunicar al Gestor de Test con el resto de gestores. Además estos mensajes son mostrados en su panel frontal en la pestaña evolución para que el programador pueda verificar el correcto funcionamiento, dispone de la posibilidad de almacenar este historial de mensajes en un fichero de texto.

Está construido con un bucle con estructuras case anidadas, Los mensajes son recibidos desde el gestor enviar y son enviados a la cola correspondiente.



Figura 107 : Gestor Mensajes

8.2.4. Gestor Enviar

Se encarga de enviar los datos al gestor mensajes para que se puedan comunicar todos los gestores.



Figura 108 : Gestor enviar

8.2.5. Gestor USB

Se encarga de todas las gestiones relacionadas con la tarjeta de relés.



Figura 109 : Gestor USB

8.2.6. Gestor Fuente de alimentación

Se encarga de gobernar la fuente de alimentación Agilent 6631B, encargada de generar la corriente inyectada a la célula para poder medir la tensión a la que es sometida.



Figura 110 : Gestor Fuente.

8.2.7. Gestor Switcher

Se encarga del control del Switcher keithley, el cual es el encargado de seleccionar una célula en cada momento y medir la tensión existente en sus bornas.

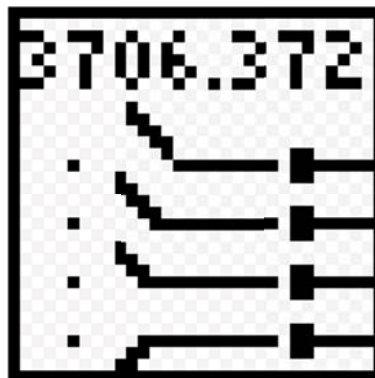


Figura 111 : Gestor switcher.

8.3. Comunicación Gestores

La comunicación entre los gestores se lleva a cabo con la programación mediante colas, mediante las cuales se envían comandos (instrucciones), para ello, como vimos en el capítulo anterior se usarán las funciones para colas de que dispone Labview.

La estructura de un gestor cualquiera quedaría como muestra la siguiente figura, en la que podemos observar la referencia global de la cola del gestor USB, como coge un elemento de la cola y lo elimina con la función dequeue, dicho elemento es uno de los casos programados dentro de la estructura case, el cual se ejecutará, y

devolverá un OK al Gestest si se ha ejecutado satisfactoriamente o un NOK en caso de que no lo fuera.

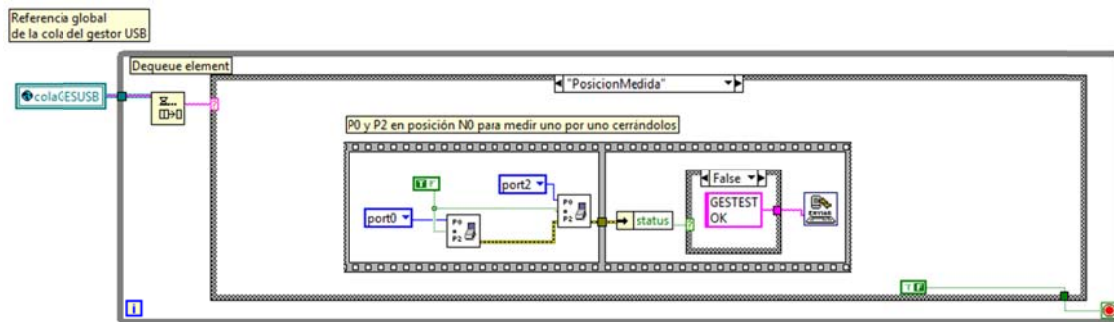


Figura 112 : Estructura Gestor.

A la hora de enviar, tenemos el vi denominado enviar, el cual añade a la cola el mensaje, y llamará a la colaGESMENSAJES el cual será el encargado de enviar el mensaje a la cola correspondiente.

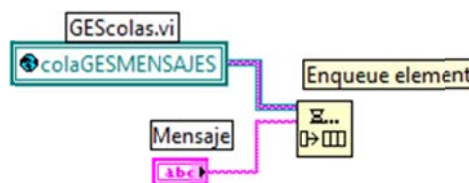


Figura 113 : Enviar.vi

Antes de enviar el mensaje o comando, es necesario definir todas las colas como variables globales, para ello estarán definidas en el GEScolas.vi



Figura 114 : GEScolas.vi

El comando enviado, sigue la siguiente estructura, la primera línea contiene el nombre del gestor al que va destinado el mensaje, la segunda línea contiene el mensaje, el cual es la opción del case que se desea ejecutar, y la tercera línea contiene la opción deseada si en el case al que estamos accediendo fuera necesario un segundo nivel de detalle.

GESTOR AL QUE SE ENVIA EL MENSAJE Mensaje Opción
--

Figura 115 : Estructura comando.

Una vez enviado el comando, abrirá el GesMensajes, el cual es el encargado de llamar al gestor correspondiente y encolarle el mensaje correspondiente para que lo pueda atender. Además el gestor de mensajes se encarga de ir mostrando todos los mensajes en el panel frontal, y a su vez archivarlos.

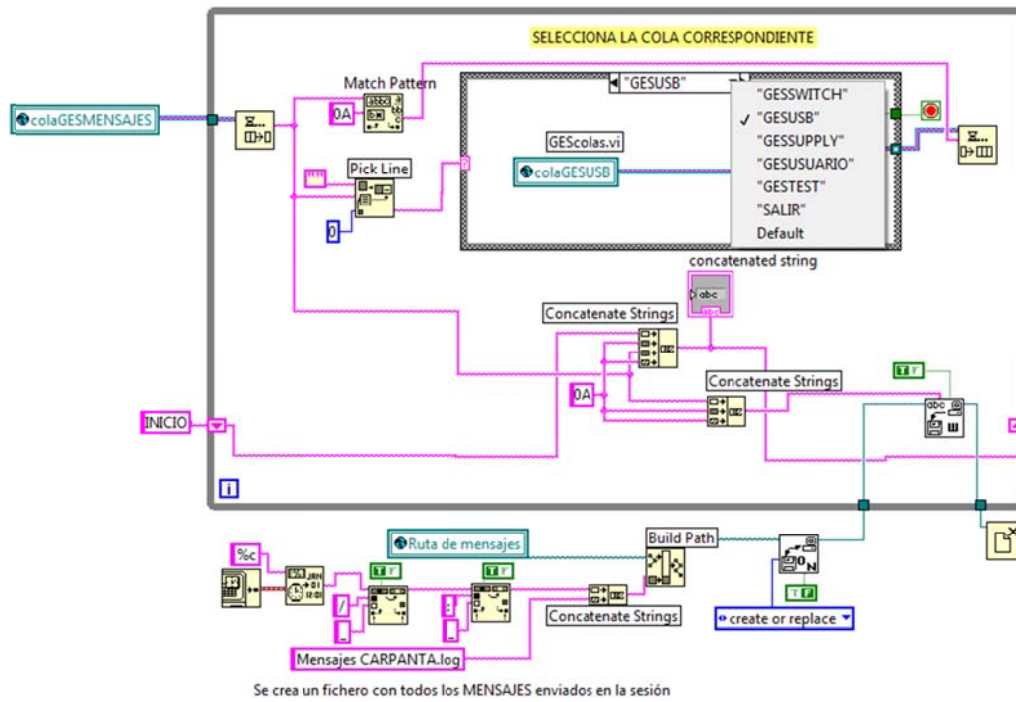


Figura 116 : Gestor Mensajes

9 Sistema de Instrumentación

En este capítulo se va a describir el programa diseñado, de una forma más minuciosa, para hacer posible una mayor comprensión del funcionamiento y uso del mismo, así como la metodología de programación usada, que ha sido descrita de forma teórica en capítulos previos, facilitando así futuras ampliaciones u otros posibles usos realizando pequeñas modificaciones de código.

9.1. Gestor Usuario

El gestor de usuario es el encargado de gestionar la interfaz con el usuario. Está formado por dos partes fundamentales: el módulo que atiende a las acciones del usuario (eventos generados por pulsaciones de teclas, clics del ratón, configuraciones de estado, etc.) y el módulo encargado de la presentación de datos en pantalla (menús, gráficos, controles, indicadores, cuadros de diálogo, etc.). Está integrado en el VI principal de la aplicación.

Por una parte tenemos el panel frontal, que es el panel de comunicación de usuario, está dividido en 4 pestañas como se puede observar.

La primera pestaña es el control de test, en dicha pestaña el usuario debe configurar las características de las medidas, de tal forma que establecerá rango de tensiones entre los que se realizarán las medidas, y el número de puntos de medidas que quiere tomar entre los valores dados, además como medida de seguridad puede configurar la limitación de corriente. En esta pantalla se puede guardar la configuración, tanto la del control de test, como la temporización y la selección de canales que veremos más adelante. En esta pantalla además se muestra información al usuario de la evolución del proceso, indicando la hora a la que se encuentra, el número de célula que se está midiendo en este momento, la última temperatura de la pt100 medida, además del tiempo que se ha sometido a estrés a las células, y el número de rondas de medida tomados.

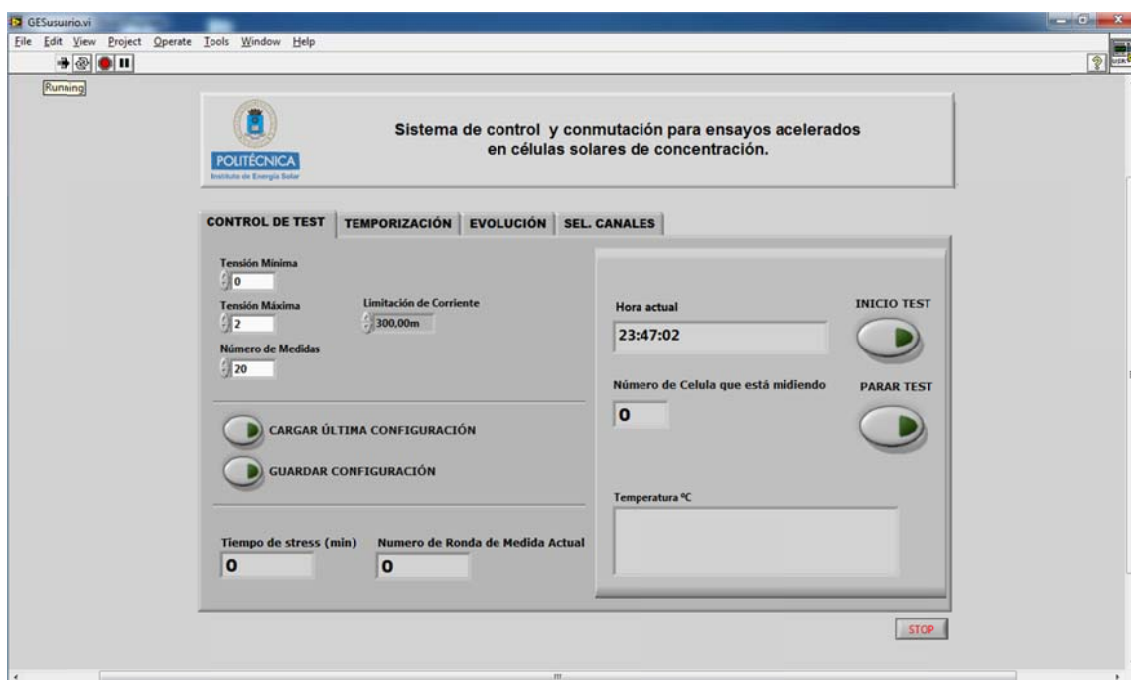


Figura 117 : Control Test Gestor Usuario

En la siguiente pestaña, temporización, el usuario debe indicar el tiempo al que van a ser sometidas las células a estrés entre medida y medida, además, se le indicará el tiempo necesario de la estabilización de la célula para poder ser medido.

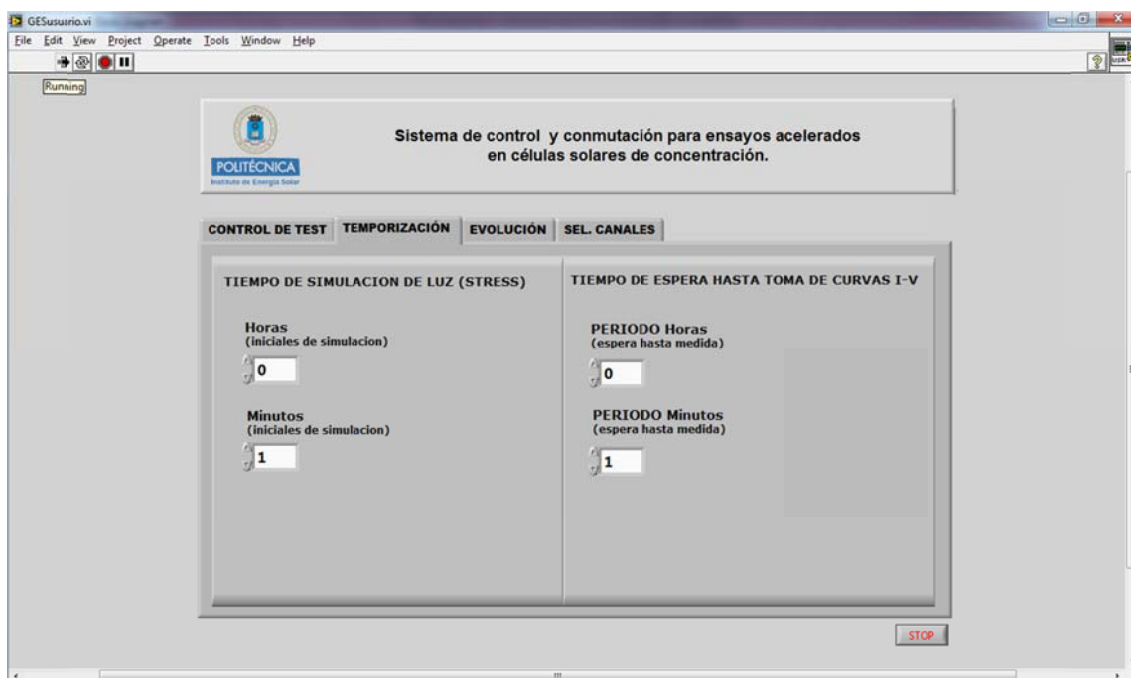


Figura 118: Temporización

En la siguiente pestaña, evolución, se muestra al usuario todo el proceso de ejecución.

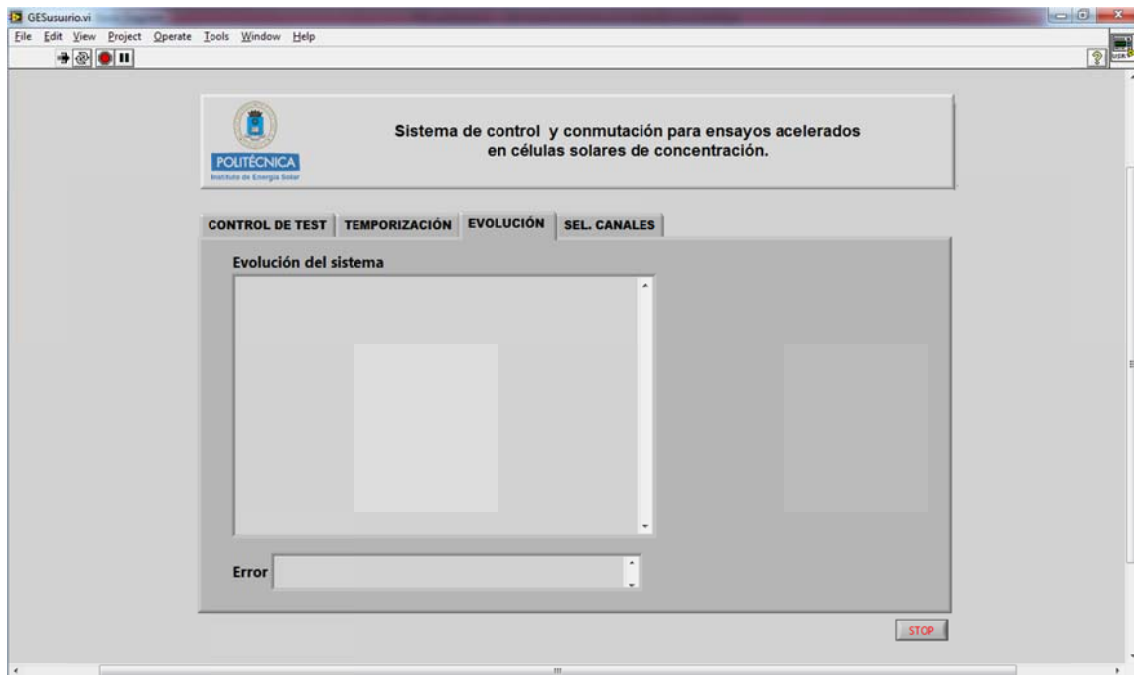


Figura 119: Evolución

Y por último, en la pestaña Sel Canales, se le pide al usuario que marque las células que desea medir, indicando los canales correspondientes, así mismo, se le indicará al usuario las tarjetas que necesita tener conectadas para poder medir en dichos canales, y le devolverá una lista como comprobante de las células seleccionadas.

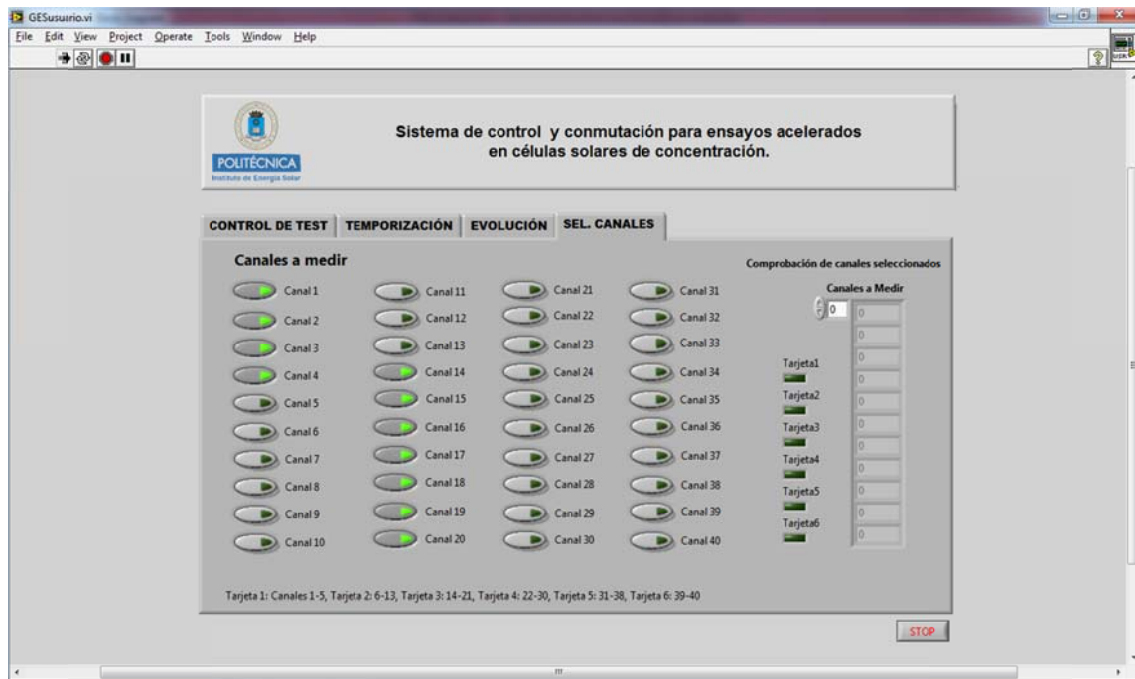


Figura 120 : Selección canales gestor usuario

En el diagrama de bloques sigue la estructura de un gestor de usuario, que tendría la siguiente forma, en la primera secuencia que podemos ver se crean las colas para todos los gestores, en la siguiente han de lanzarse todos los gestores a los que hace referencia la aplicación, constará de un bucle while con la atención de eventos de usuario, y por último un módulo gestor de usuario de presentación de datos en pantalla.

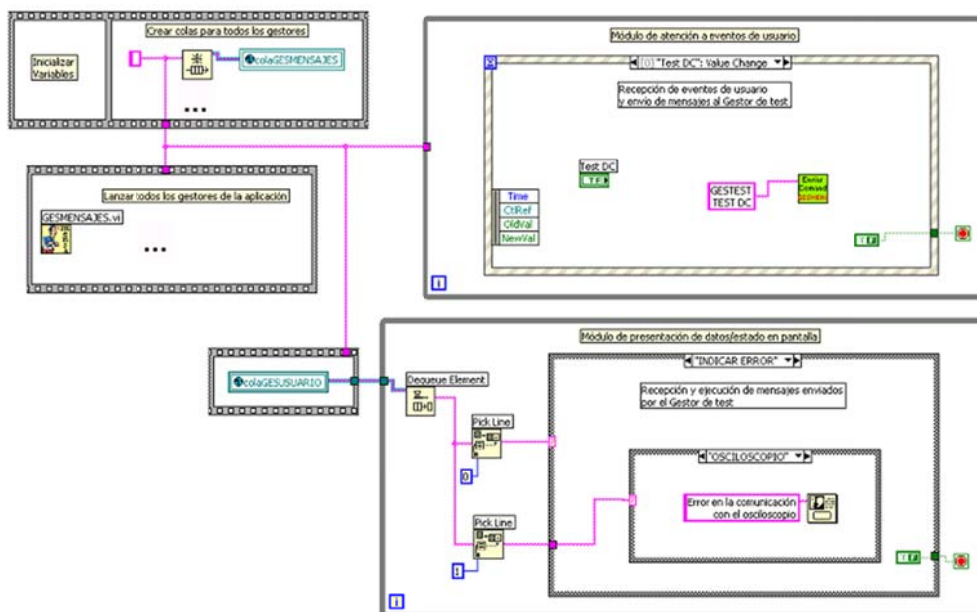


Figura 121: Modelo gestor usuario.

A continuación, pasaremos a describir las partes mencionadas dentro del programa de toma ensayos acelerados para células solares.

La primera secuencia incluye la creación de colas para los gestores, y la inicialización de todas las variables, esto se ejecuta al arrancar el programa LabView (antes de pulsar Iniciar), introduce un valor inicial de seguridad a todas las variables que deben ser configuradas por el usuario, además resetea los valores que muestra al usuario para no mostrar valores anteriores, y lee la configuración de los canales seleccionado, además se asegura que las fuentes de corriente estén apagadas en su inicio.

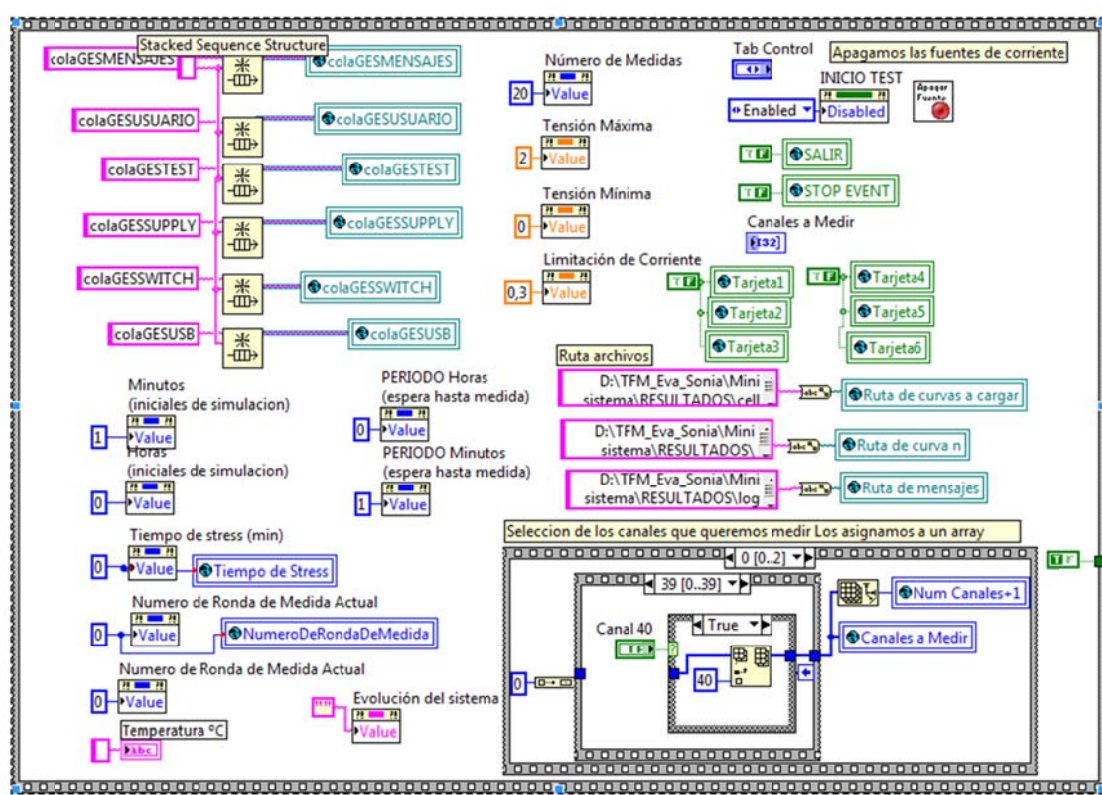


Figura 122: Inicialización colas

La segunda secuencia se encarga de arrancar todos los gestores que se describirán en los siguientes apartados.



Figura 123 : Inicia Gestores

Se ha creado una tercera secuencia que si bien no aparecía en el esquema inicial, pero se ha considerado necesaria su inclusión para mostrar al usuario la hora en la que se encuentra para que pueda verificar los tiempos de estrés si lo creyera conveniente, y actualizar la variable de la célula que se está midiendo, sin necesidad de pasar por el estado de actualización que se especificará más adelante.

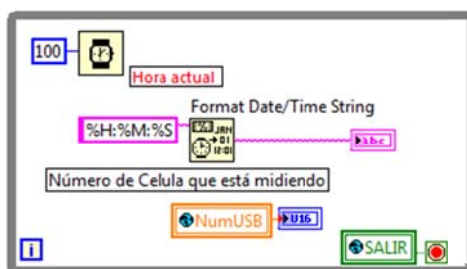


Figura 124 : Control horario

La siguiente secuencia, corresponde a otro bucle while con un gestor de eventos, el cual recoge los eventos generados por el usuario, estos eventos pueden ser INICIO, STOP, CARGAR CONFIGURACIÓN o GUARDAR CONFIGURACIÓN.

Si el usuario pulsa INICIO, se manda la orden al Gestest de INICIAR, y además se deshabilita el botón para que el usuario no lo vuelva a pulsar. Si pulsara el botón STOP se envía la orden de parar toda la ejecución, y apagar las fuentes de corriente.

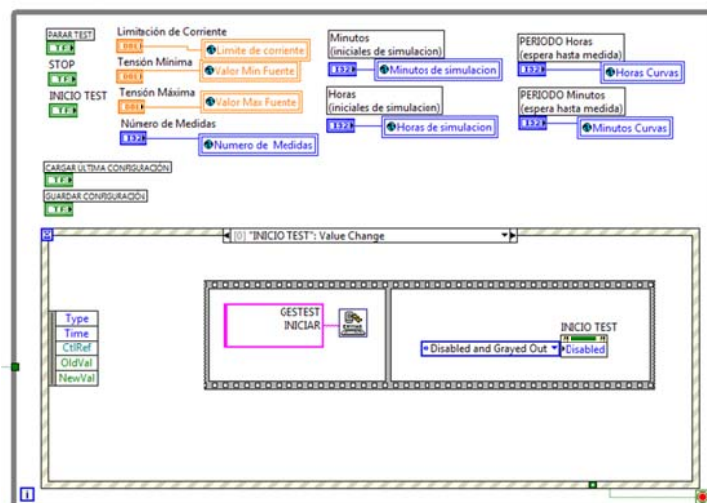


Figura 125 : Gestor eventos- Inicio.

Si pulsa cargar última configuración, automáticamente rellenará todos los datos solicitados al usuario con los valores previamente almacenados.

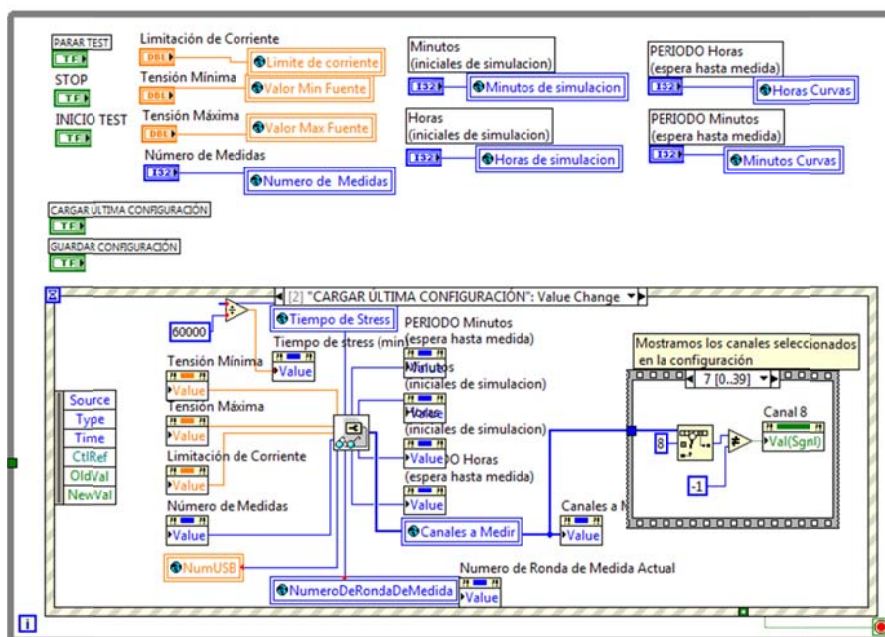


Figura 126: Gestor eventos- Cargar.

Y por último guardar configuración:

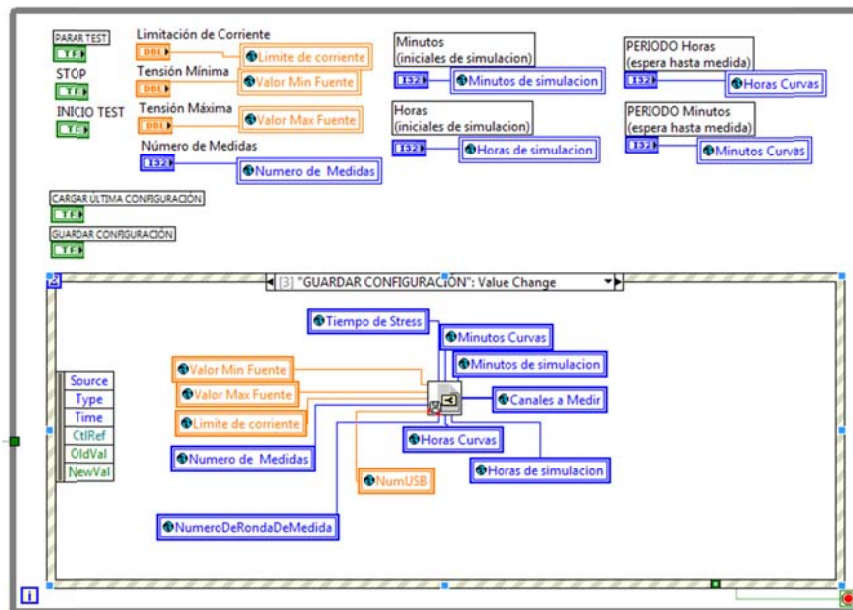


Figura 127: Gestor eventos: Guardar.

En el último frame o estado es el gestor de usuario, el encargado de mostrar la evolución del programa al usuario, se compone de diferentes casos dependiendo de la llamada realizada, el primero de ellos, denominado evolución, muestra en la pantalla al usuario el mensaje enviado a la cola por los distintos gestores.

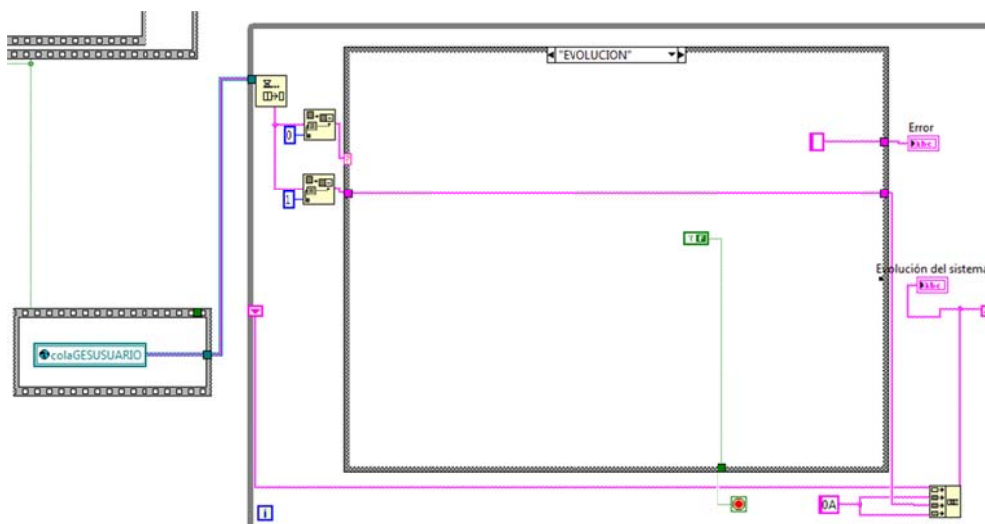


Figura 128: Evolución.

Otro caso, es el de actualización, el cual se encarga de comprobar si el usuario ha cambiado alguno de los valores configurados inicialmente, entre dichos cambios, podría modificar la selección de canales, si hubiera una célula que hubiera dejado de

funcionar, no es necesario estarla tomando medidas innecesariamente. Y el tiempo de estrés y de espera antes de proceder a realizar las medidas.

Además en este estado se actualiza la información mostrada al usuario, como es el tiempo de estrés acumulado por la célula, y el número de la última ronda de medida realizada.

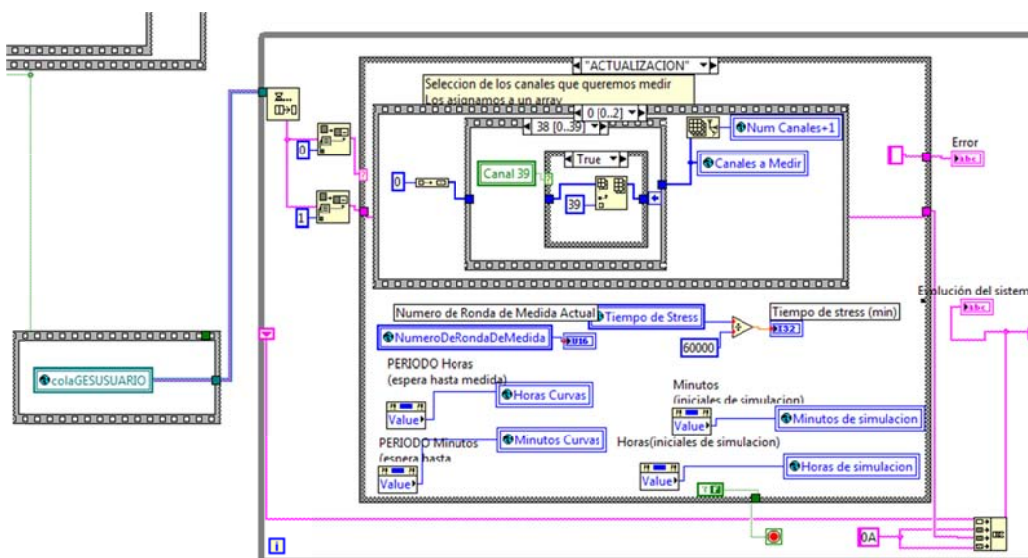


Figura 129: Actualización.

Y por último, el estado error, el cual se encarga de mostrar al usuario que ha habido un error, leerá de la cola el dispositivo que ha generado el error y lo mostrará al usuario, se apagarán las fuentes de corriente, y se ordenará parar la ejecución del programa.

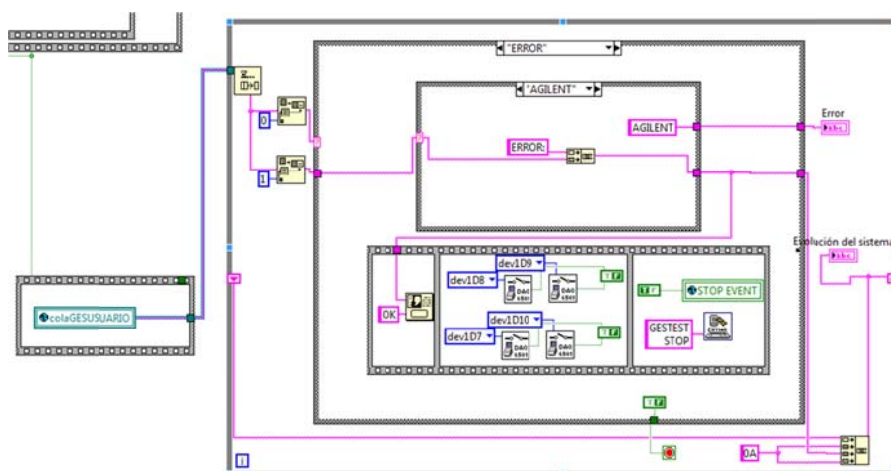


Figura 130 Error.

9.2. Estados Gestest

9.2.1. Funcionamiento del Gestor de Test

El gestor de test es el encargado de controlar el funcionamiento global de la aplicación. Está implementado como un autómata de estados y se encarga de enviar órdenes al resto de los gestores, indicando las acciones que en cada momento éstos deben realizar. Así mismo, el Gestor de test recibe del resto de los gestores información sobre los resultados de la ejecución de las acciones realizadas por éstos de tal forma que estos devolverán un OK, si el resultado de la operación ha sido satisfactorio y sin problemas, el cual es el indicador que el autómata pasa a ejecutar su siguiente estado, o pueden devolver un NOK, si hubiera habido algún problema o error al realizar la operación solicitada por el Gestor de Test, en este caso, el gestor de Test entrará en el estado configurado como NOK, en el que se enviará al Gestor de Usuario un error, como primera línea de comando, y como segunda línea de comando el dispositivo que ha mandado el mensaje de error al Gestor de Test. Dicho gestor de usuario, le mostrará un aviso en el panel de Evolución y parará la ejecución para que el usuario pueda solventar el error dado.

La siguiente figura muestra la máquina de estados diseñada para poder llevar a cabo el ensayo sobre las células solares, en cada uno de ellos el gestor de test se comunicará con el gestor correspondiente, en dicho diagrama solo se muestra el funcionamiento en caso de que la respuesta del Gestor correspondiente sea OK, para poder realizar el autómata al completo, si en cualquier de los estados devolviera un NOK, daría el error correspondiente y se iría al estado de reposo.

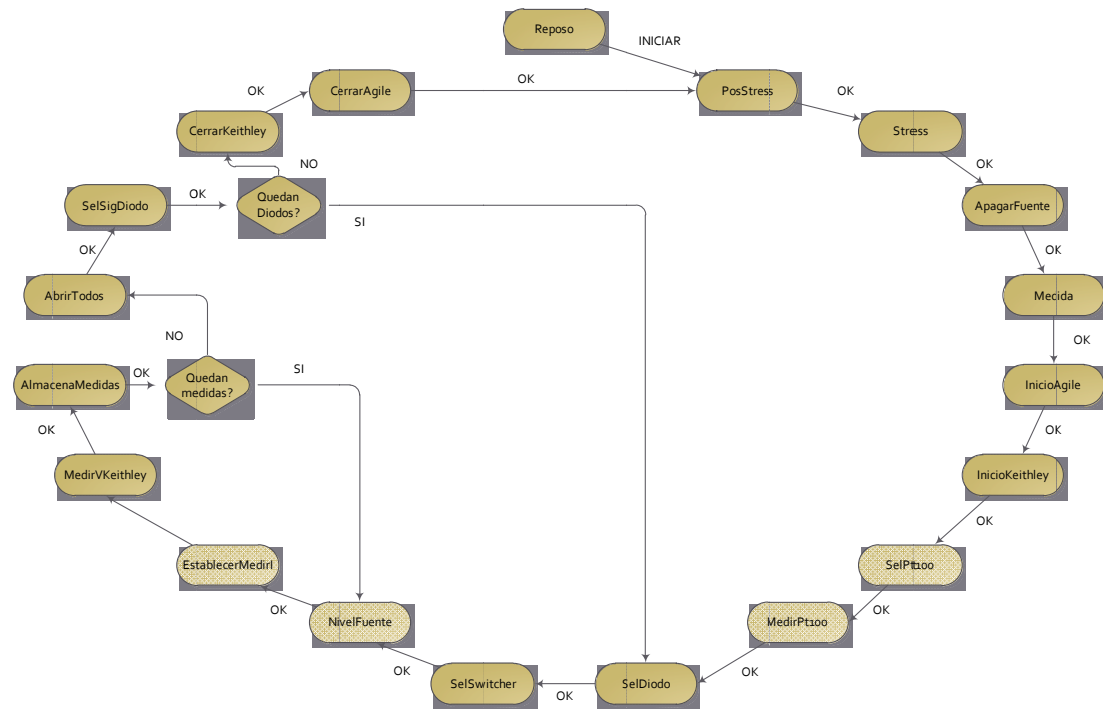


Figura 131: Diagrama de estados GesTest.

Para el diseño de la máquina de estados se ha llevado a cabo la siguiente estructura de colas.

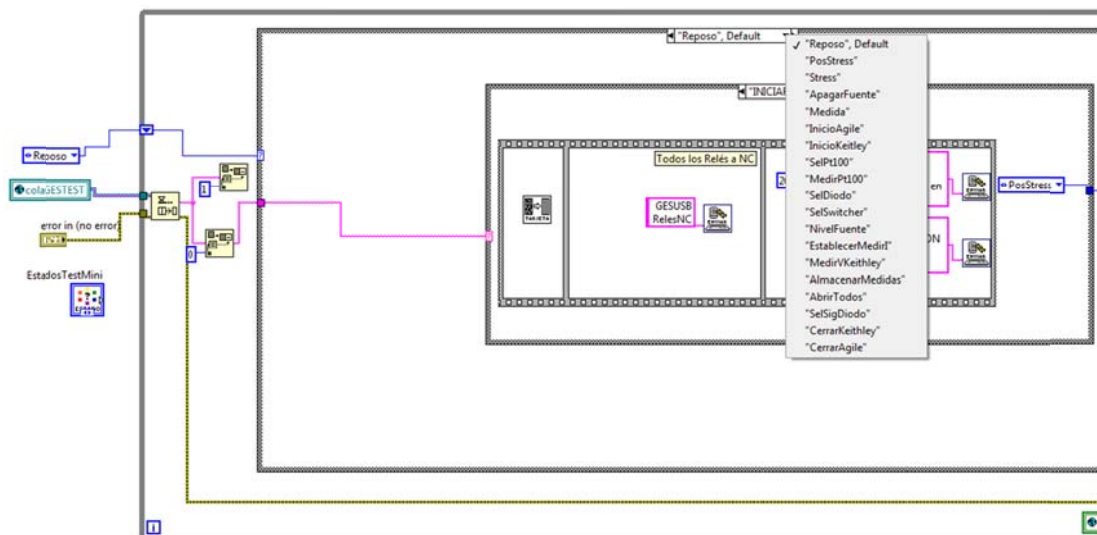


Figura 132: Estados.

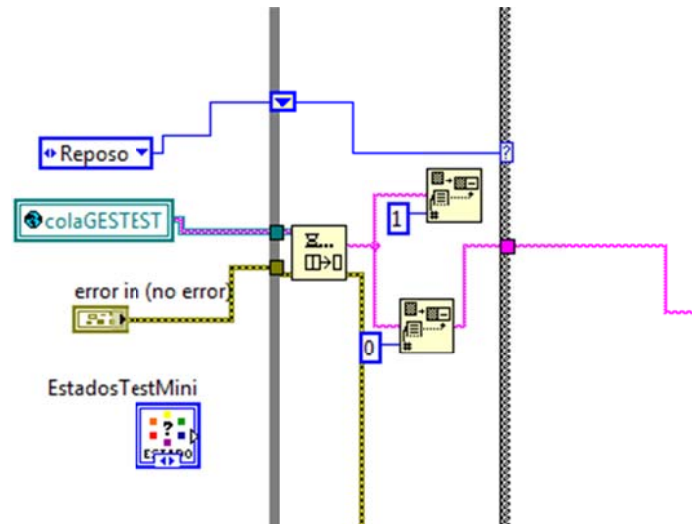


Figura 133: Detalle Colas.

9.2.2. Reposo

Una vez ejecutado nuestro programa permanecerá en el estado de reposo hasta que el usuario una vez echas todas las configuraciones pulse la opción INICIAR.

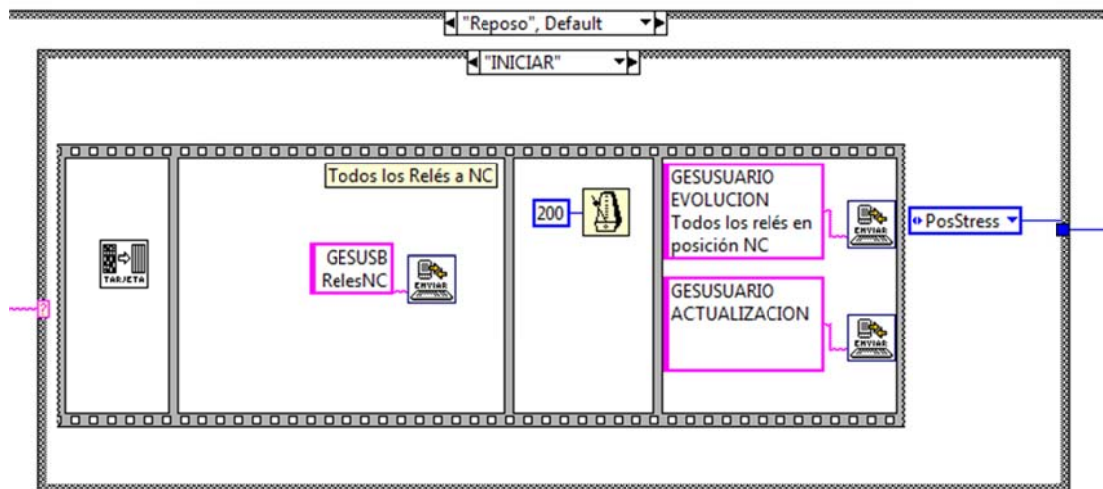


Figura 134 : Reposo-> Iniciar

Una vez pulsado nuestro botón iniciar, entraremos en el estado Iniciar, lo primero que hará dicho estado, es indicar al usuario el número de tarjetas que debe tener conectadas en función del número de canales seleccionados, para ello, se llamará a NumTarjetas.vi , una vez seleccionado, se seleccionan todos los relés en posición NC, de tal forma que no están sometidos a ninguna actividad. Actualizaremos el estado del proceso mediante EVOLUCIÓN para que el usuario

pueda seguir el proceso en todo momento, y pasaremos al siguiente estado, PosStress.

9.2.3. PosStress

En este estado, se indica al usuario que va a comenzar a someter a estrés a las células, para ello, selecciona todos los relés en la posición stress COM-NO. Y pasamos a estresar las células.

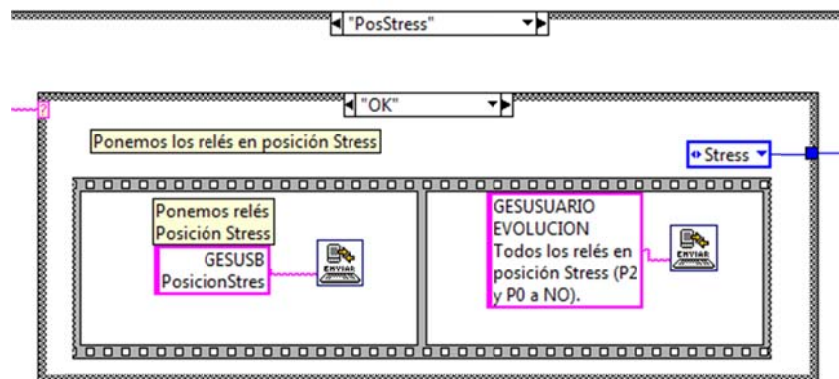


Figura 135 : PosStress.

9.2.4. Stress

En este estado encenderemos la fuente de corriente, para ello, enviaremos el comando EncenderFuenteCorriente al Gestor del USB. A partir de este momento, el tiempo de estrés comenzará, por lo que comenzará la primera ronda de medida, los datos seleccionados serán los vigentes durante esta ronda, por lo que el programa procede a almacenarlos en un archivo. En la siguiente secuencia, se encuentra el tiempo de espera que previamente el usuario ha seleccionado, mientras dicho tiempo no finalice o no se pulse la opción salir, se mantendrá en dicha secuencia. Una vez finalizado, si hubiera finalizado por finalización del tiempo, pasaremos al siguiente estado, ApagarFuente, e incrementaremos en 1 nuestra ronda de medida. Si hubiera finalizado porque el usuario haya pulsado salir, pasará al estado de reposo.

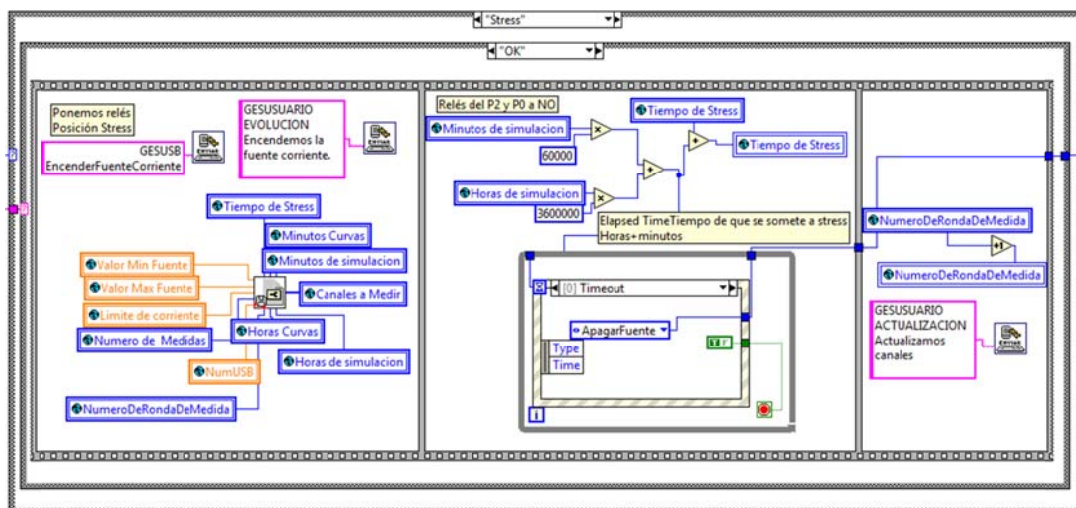


Figura 136 : Stress

9.2.5. ApagarFuente

Como su propio nombre indica, en este estado se ordenará al Gestor USB que apague la fuente de corriente para dejar de someter a estrés a las células. Una vez apagada, se pasará al siguiente estado, en cual se comenzará a medir.

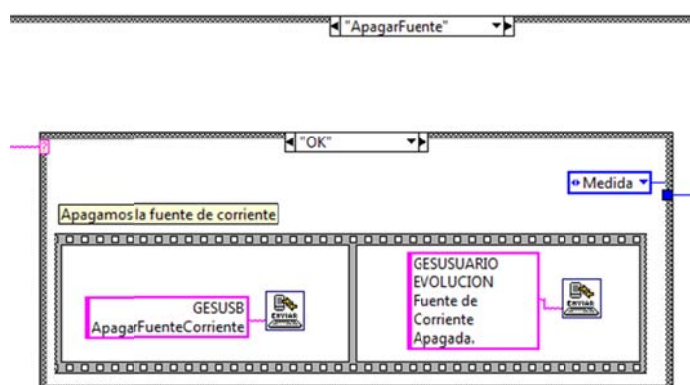


Figura 137 : Apagar fuente de corriente

9.2.6. Medida

En este estado, lo primero que se comprueba es que el usuario ha seleccionado algún canal, para ello, se comprueba que el número de canales no es igual a 1, si lo fuera, indica que no ha seleccionado nada y se le indica al usuario que debe seleccionar los relés que desea medir para poder realizar las medidas.

Si es distinto de 1, se procede a tomar las medidas, para ello, se usará la variable recorre canal, la cual indica la posición que vamos a leer de la tabla (canales a medir) que contiene los relés seleccionados. Seleccionamos todos los relés en posición NO, para asegurarnos que ninguna de ellos está conectado al Agile para ser medido, realizamos un tiempo de espera de estabilización para asegurarnos que las células se han estabilizado en temperatura y no tienen nada de estrés, una vez finalizado dicho tiempo se procede a tomar medidas, pasando al estado inicioAgile.

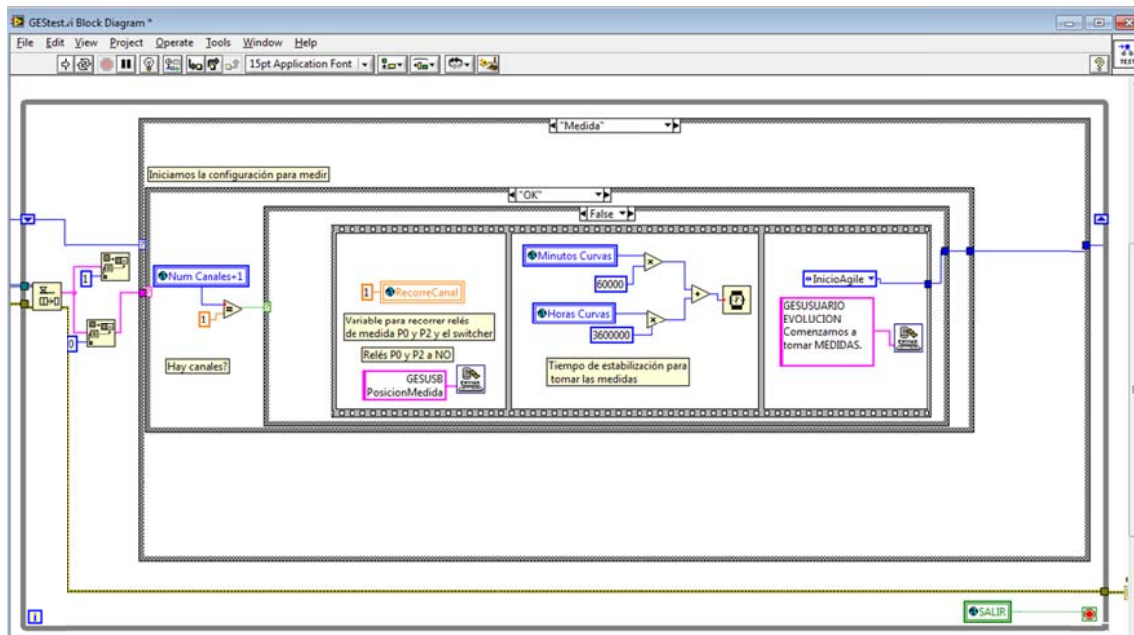


Figura 138: Medida.

9.2.7. InicioAgile

Le envía al Gestor de la Fuente de Alimentación, Agilent, el comando para que inicie.



Figura 139 : InicioAgile

9.2.8. InicioKeithley

Le envía al Switcher, Keithley, el comando para que inicie.

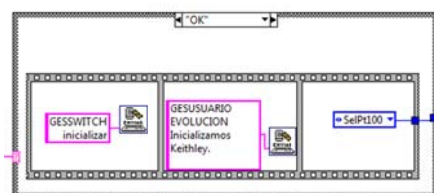


Figura 140: Inicio Keithley

9.2.9. SelPt100

Seleccionamos el sensor de temperatura que previamente se ha introducido en el Horno, y se ha conectado en un canal del switcher keithley, como se puede observar, dicho canal es el número 40, por lo que la célula número 40 estaría inutilizada en este ensayo, puesto que se inicia con 20 células se ha optado por colocarla en este canal. En la siguiente secuencia, se cierra el canal seleccionado.

Cada vez que se realicen medidas se almacena el valor del sensor de temperatura que indicará la temperatura real de la célula.

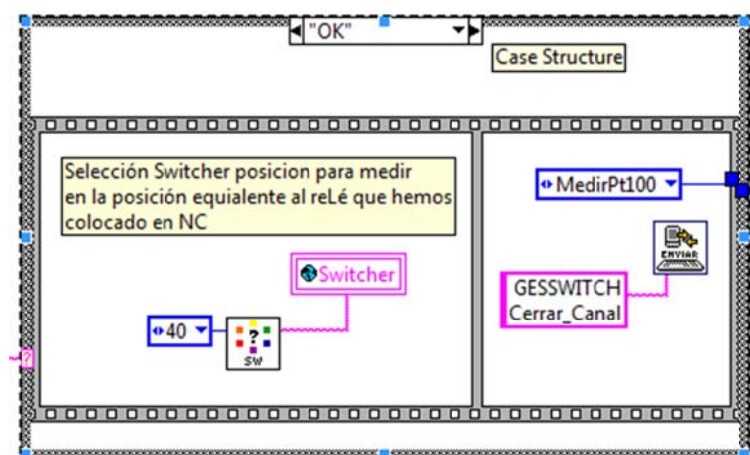


Figura 141: Seleccionar Pt100

9.2.10. Medir Pt100

En este estado se llamar al Gestor del Switcher para indicarle que debe medir la temperatura y almacenarla, en este caso se han programado 2 opciones, en la que se ve en la imagen MedirTermocouple, se medirá la temperatura de un termopar, por lo que en el canal 40 que tenemos seleccionado, tendría que haber conectado un termopar. Si lo que tenemos conectado en el canal 40 es una termoresistencia RTD,

más conocía como Pt100, lo que dió nombre al estado en el que nos encontramos, habrá que modificar “MedirTermocouple” por “Medir R”.



Figura 142: Medir temperatura

A partir de este punto es necesario esquematizar el punto de la máquina de estados en el que nos encontramos, explicando una pequeña máquina de estados generada dentro de la general para comprender mejor su funcionamiento, dicha máquina de estados comienza con la selección de célula o diodo correspondiente de las seleccionadas por el usuario, una vez seleccionado, se cerrará el canal correspondiente, se tomarán las medidas, se abrirá el canal cerrado, y si hay más células seleccionadas por el usuario, volver a comenzar el proceso.

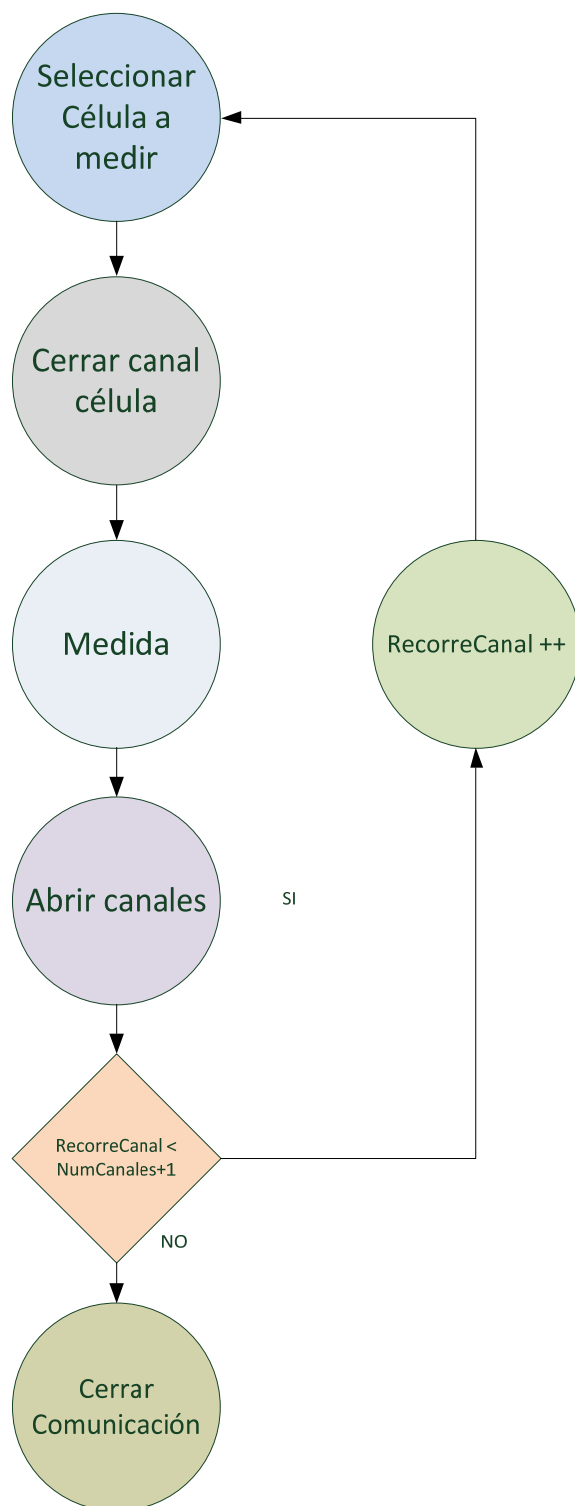


Figura 143 : Máquina estados selección célula

9.2.11. SelDiodo

En este estado, seleccionaremos la célula con la que vamos a empezar a tomar medidas, para ello, necesitaremos las variables globales, canales a medir, que es un array que almacena las células que ha seleccionado el usuario para medir, RecorreCanal que hace los efectos de contador para saber qué posición de la tabla estamos leyendo, y NumUSB que es el número de la célula que vamos a medir, dicho número está asociado a una pareja de relés gobernados por el USB, el cual en el siguiente frame se ordena colocarlos en la posición medir. Y también estará asociado a un canal del switcher que deberá seleccionar para poder medir, esto se realizará en el siguiente estado, puesto que hay que controlar la comunicación con todos los dispositivos.

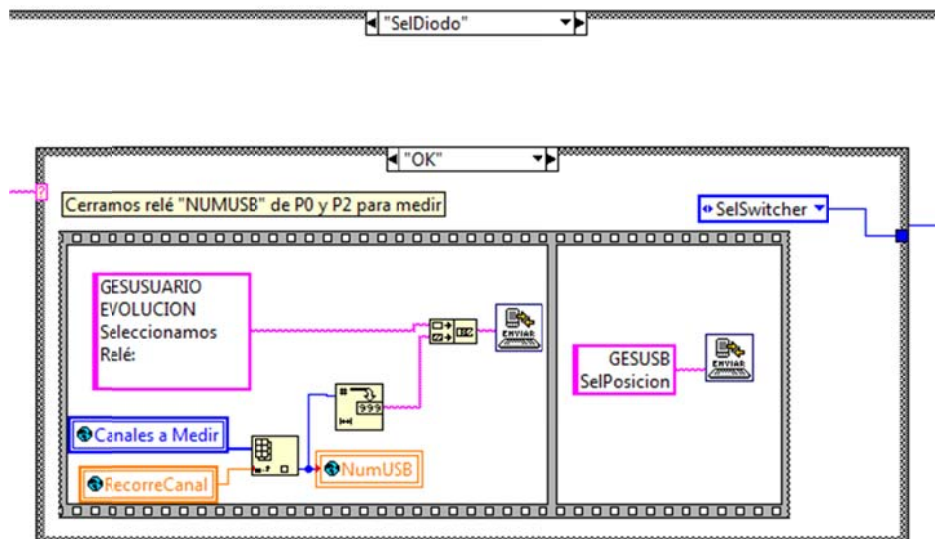


Figura 144 : SelDiodo

9.2.12. SelSwitcher

En este estado, selecciona la posición del switcher keithley asociada a la célula que queremos medir, para ello, usamos la variable global NumUSB que indica la posición que estamos leyendo en este momento, y llamaría a ControlEstadosSwitcher.vi (ver: Figura 188 : ControlEstadosSwitcher.vi), al cual se le pasa como parámetro el canal que está leyendo y devuelve una cadena con el número que hay que pasarle al switcher, que está compuesto de la tarjeta que esté conectada en nuestro caso la primera y el canal en centenas seguido de 1911 que indica que estamos usando la tarjeta 1 para medir a 2 hilos en los canales comprendidos del 1 al 20, si se está usando un canal comprendido entre el 21 y el 40 nos devolverá el canal

seguido de 1921, dicha cadena se guarda en la variable global switcher para posteriormente ser leída. Por ejemplo, si queremos medir el canal 33, Switcher tendrá almacenado “1033,1921”. La medida se hace a dos hilos con el keithley y para hacer la medida a 4 hilos se utiliza la segunda medida con el dispositivo Agilent.

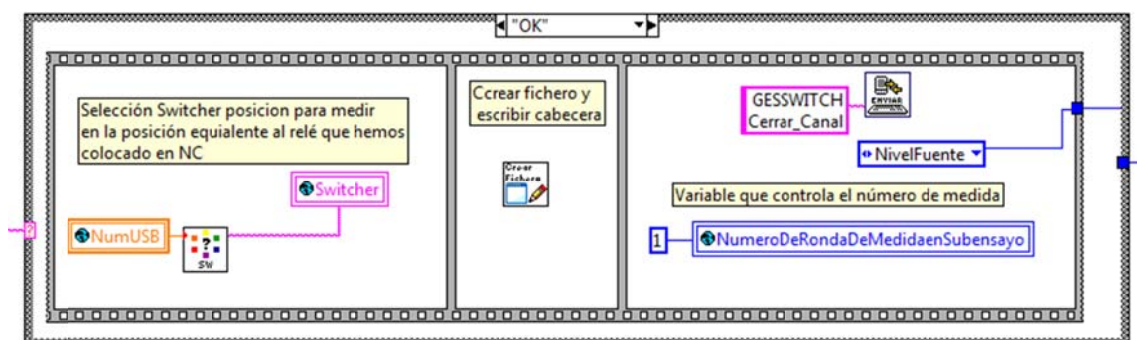


Figura 145 : SelSwitcher

A continuación en el siguiente frame se llama a la función `escribir_cabecera.vi`, dentro del cual a su vez ejecuta `guardar_cabecera.vi`, crea o abre el archivo y almacena las cabeceras o textos fijos para posteriormente almacenar los datos.

Y en el último frame, llama al GESWITCH para que cierre los canales seleccionados, además inicializa el valor de la variable global `NumeroDeRondaDeMedidaSubensayo`, variable usada como contador del número de medidas realizadas a la célula para compararla con el número de medidas solicitadas por el usuario, además esta variable se usará para establecer la tensión correspondiente con el agilent, como un valor de la ecuación que calculará la tensión, indicando en número de medida en el que nos encontramos, como se verá en el siguiente punto.

Una vez seleccionado el canal del switcher correspondiente a la célula que se desea medir, comenzará una segunda pequeña máquina de estados dentro de la última descrita, como desarrollo del estado medida (ver: *Figura 143 : Máquina estados selección célula*), para una mayor comprensión se añade un diagrama de funcionamiento que incluye los siguientes estados que serán descritos, en el que se puede observar que para cada célula hay que realizar un número de medidas indeterminado, por lo que se hace un bucle en el que se establece la tensión en la célula, se mide tensión y corriente, y se comprueba si se han realizado todas las medidas, si no se han completado, se incrementa en uno la variable que contiene el

número de medidas que se han realizado y se vuelve a establecer la nueva tensión a la célula.

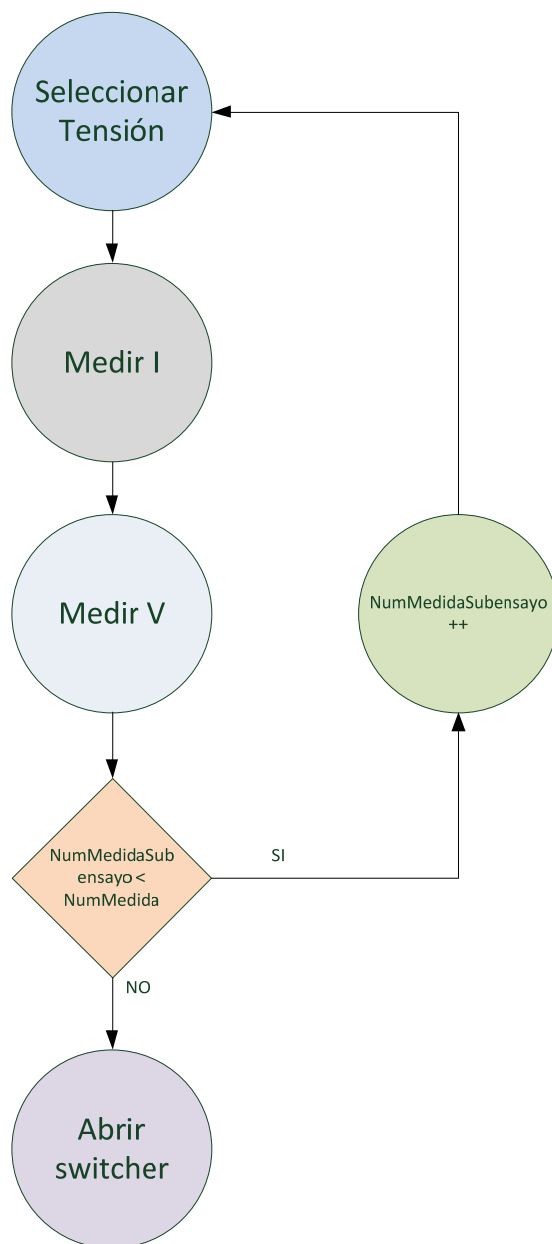


Figura 146 : Diagrama medida curva IV

9.2.13. NivelFuente

En este estado, se establece el valor de tensión que ha de ponerse en la fuente de alimentación Agilent, dicho valor se obtiene de los parámetros introducidos por el usuario, según la ecuación:

$$V = \left\{ \left[\frac{(V_{Max} - V_{min})}{\text{NumeroMedidas}} \right] \times \text{NumRonda} \right\} + V_{min}$$

Dónde V_{max} es la tensión máxima, V_{min} la tensión mínima, Número de medidas el número de muestras total que desea tomar el usuario, y NumRonda, el número de la medida en la que se encuentra actualmente.

Una vez resuelta la ecuación, almacena el valor en la variable global NivelFuente, y envía el comando al Gestor de la Fuente Nivel_Fuente.

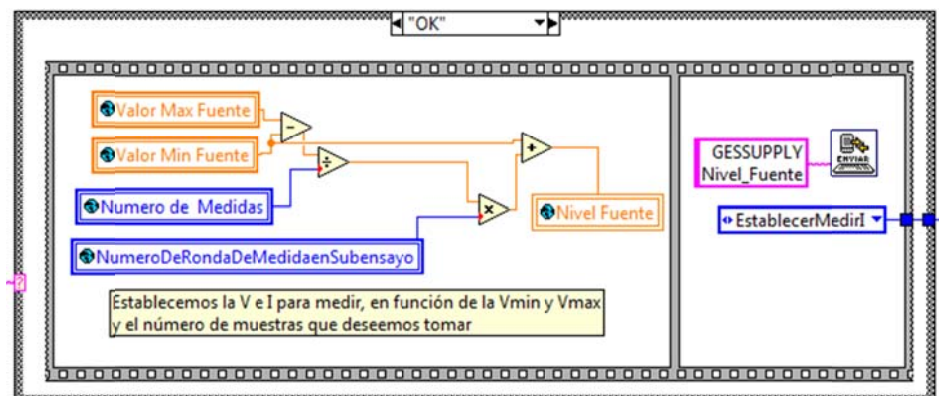


Figura 147 : Nivel Fuente

Para una mayor accesibilidad a posibles cambios, esta ecuación se implementa en un nuevo subprograma minMAX.vi.

9.2.14. EstablecerMedirI

Envía el comando Medir a la Fuente de Alimentación Agilent, la cual se encargará de medir con la corriente que llega a la célula para la primera medida a 4 hilos.

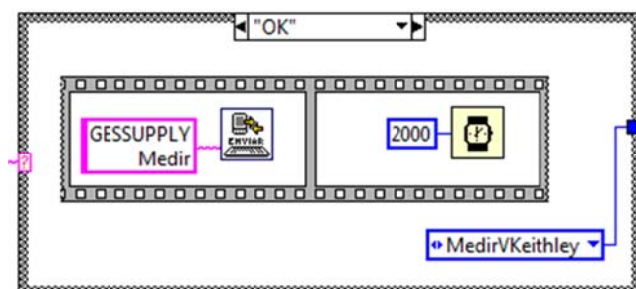


Figura 148 : EsablecerMedirI

9.2.15. MedirVKeithley

En este estado se envía el comando Medir V para que el Switcher Keithley se encargue de medir la tensión en bornas de la célula, completando así la medida a 4 hilos.

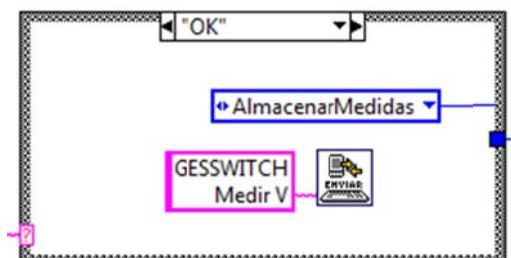


Figura 149 : MedirVKeithley

9.2.16. AlmacenarMedidas

En este estado se almacena en el archivo creado anteriormente el valor de tensión y de corriente que se acaba de medir. Se comprueba si se han realizado todas las medidas solicitadas por el usuario, comparando el número de medida en subensayo en que se encuentra con el número de medidas solicitadas, si este fuera menor, se incrementaría en 1 el número de medida en el subensayo y volvería a establecer el siguiente nivel fuente, haciendo el cálculo con el nuevo valor de número de ronda de medida en subensayo.

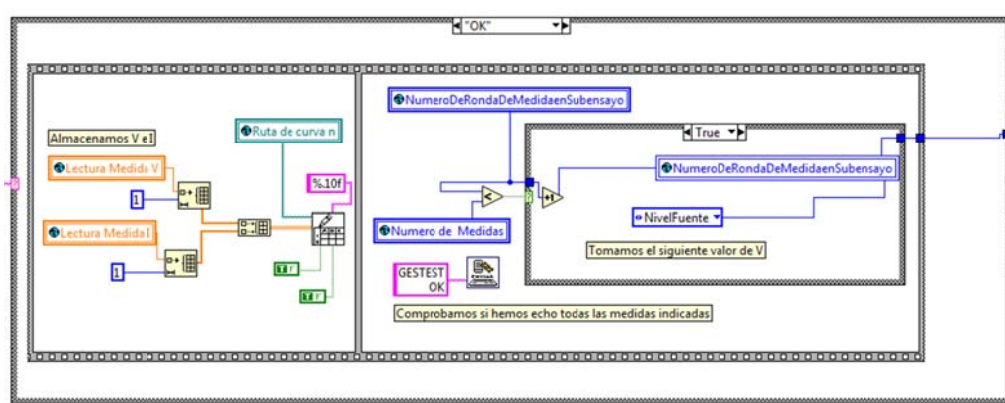


Figura 150: Almacenar medidas.

9.2.17. Abrir todos

En este estado ya se ha terminado de medir la célula en cuestión y se abren los canales que previamente se habían cerrado para llevar a cabo las mediciones.



Figura 151: Abrir todos

9.2.18. SelSigDiodo

En este estado se conmutan los relés que estaban siendo usados para medir a la posición de reposo, y se selecciona la siguiente célula a medir, para ello previamente se comprueba si la posición del canal por la que se encuentra tomando medidas es menor al número de canales que el usuario desea medir, de ser así, se pasa a seleccionar la siguiente célula incrementando la variable recorre canal en una posición, y volviendo al estado Sel Diodo.

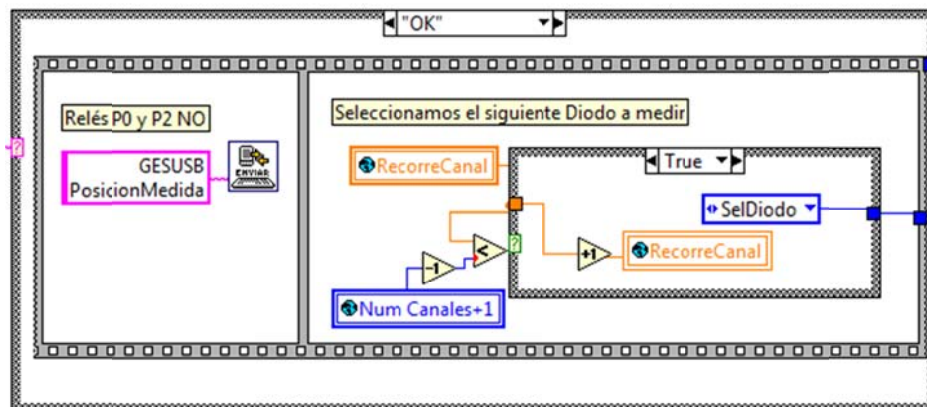


Figura 152 : SelSigDiodo

Si se hubiera terminado de medir todas las células que el usuario ha seleccionado, habría terminado el proceso de medida y estaría listo para volver al estado de estrés. Para ello, será necesario cerrar la comunicación con los dispositivos de medida.

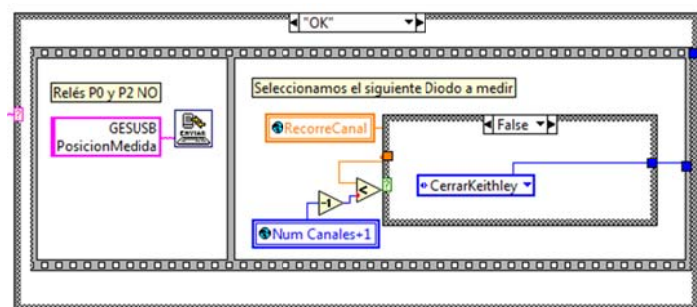


Figura 153: Sel Sig Diodo fin.

9.2.19. CerrarKeithley

Finaliza la comunicación del dispositivo Keithley, liberándolo, hasta que se vuelva a completar el estado de estrés y de espera y vuelva a ser necesario realizar medidas.

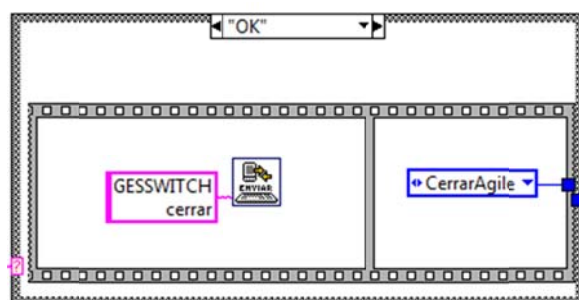


Figura 154: Cerrar Keithley

9.2.20. CerrarAgile

Finaliza la comunicación con el dispositivo Agilent, liberándolo, hasta que se vuelva a completar el estado de estrés y de espera y vuelva a ser necesario realizar medidas.

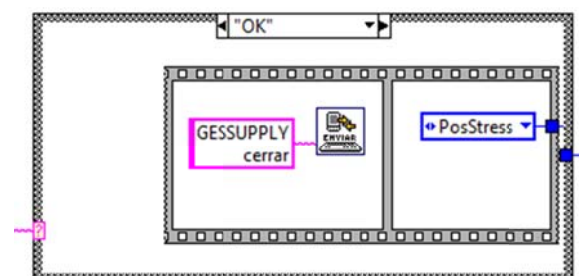


Figura 155: Cerrar Agilent

Una vez completado este paso, se volverá a ejecutar de nuevo la máquina de estado volviendo a la PosStres.

9.3. Gestor USB

Este gestor se encarga de realizar todas las tareas relacionadas con las tarjetas de relés, las cuales están controladas con el dispositivo de National Instrument DAQ USB.

Este gestor es el encargado de conmutar los relés del enchufe de alimentación, los cuales se encargan de que las fuentes de corriente sólo estén encendidas cuando es necesario para someter a estrés a la célula, además en este gestor se puede controlar cuando la célula está siendo sometida a estrés o va a ser medida.

A continuación se detallan los estados de que dispone este gestor.

9.3.1. PosicionStress

Este estado se modifica la posición de todos los relés, tanto el puerto 2 como el puerto 0, situándolos en la posición NO, todos salvo los relés tratados de forma independiente de la primera tarjeta seleccionada, y los sobrantes de la sexta si la hubiera. Para ello, se llama al subVi [usb_all.vi](#), éste devolverá un ok si todo ha sido correcto, que se le enviará al Gestest para que continúe con la ejecución del programa.

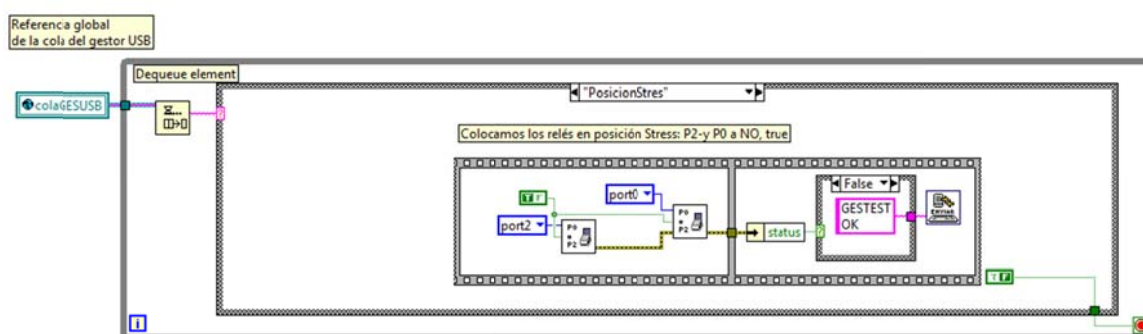


Figura 156 : Posicion Stress

9.3.2. SelPosicion

En este estado se selecciona los relés adyacentes a la célula que queremos medir, el número de célula está contenido en la variable NumUSB, como ya se ha

mencionado anteriormente, dicho valor se le pasará los VI creados para seleccionar su canal equivalente en el puerto 0 y en el puerto 2 (*ControlEstadosPuertos.vi* y *ControlEstadosPuertos2.vi*), posteriormente pasarán a situarse en la posición NC para poder ser medidos, de esto se encargará el VI *usb_cell.vi*. Si la operación se ha completado satisfactoriamente devolverá un OK.

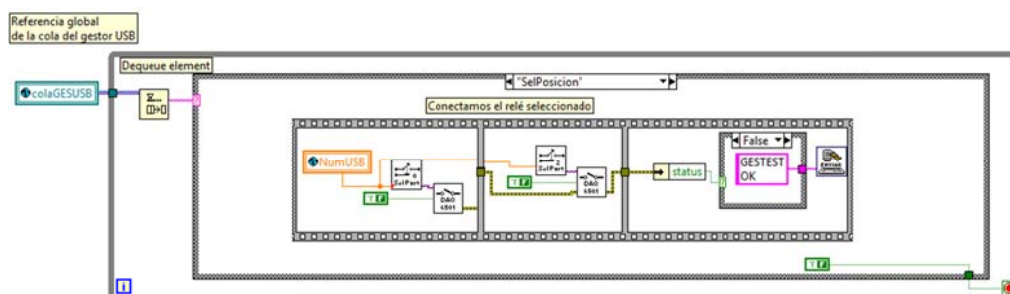


Figura 157: SelPosicion

9.3.3. Relés NC

Conmuta todos los relés a la posición NC, esta opción en el diseño final no se usa, pero se ha preferido dejarla por si en futuras mejoras fuera necesaria.

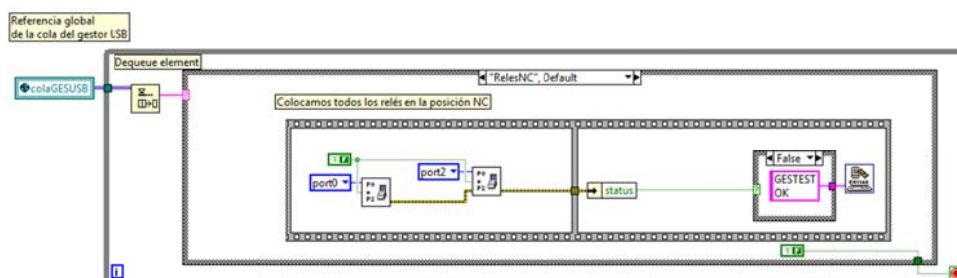


Figura 158: Relés en posición NC.

9.3.4. ApagarFuenteCorriente

Este estado se encarga de apagar las fuentes de corriente, para ello, conmuta los relés de forma que no lleguen los 220V de la alimentación a las fuentes de corriente.

En la programación se ha optado por conectar el relé 8 a NC y el relé 7 a NO, como se puede apreciar en el esquema abajo, con eso sería necesario, pero para asegurarse de que si fallara alguno, seguirían sin estar alimentadas las fuentes de

corriente, se conmuta el relé número 9 en la posición NO y el relé número 10 en la posición NC.

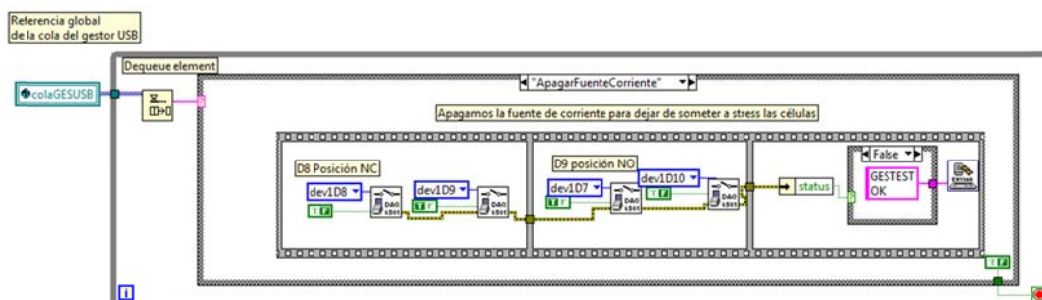


Figura 159: Apagar fuente de corriente

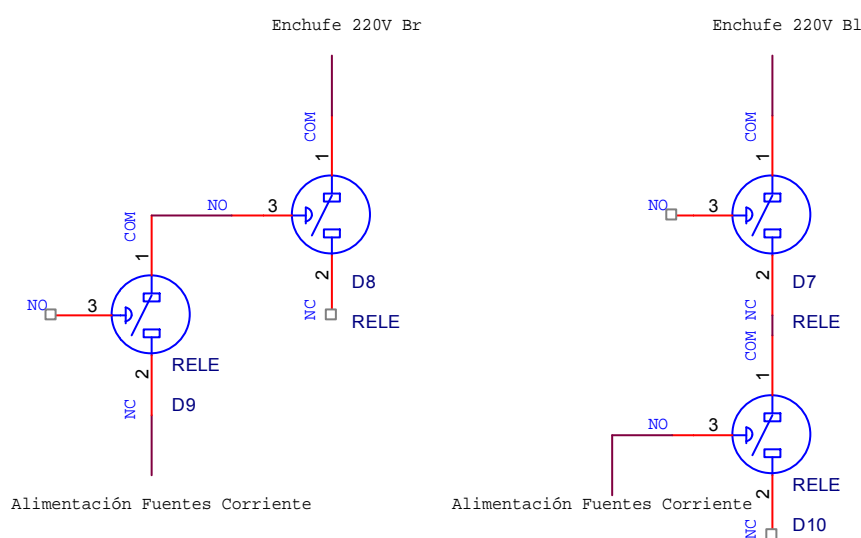


Figura 160: Esquema conexión fuente corriente.

9.3.5. EncenderFuenteCorriente

En este estado se controlan los relés encargados de encender la fuente de corriente, para dar una mayor seguridad a la instalación, se ha optado por cada cable de alimentación del enchufe, pasarle por dos relés, de tal forma que sólo va a estar encendida cuando esté gobernada por el programa.

Para encender la fuente de corriente, como se puede ver en el esquema hardware y en el diseño software que se muestran a continuación, es necesario conmutar el relé número 8 en la posición NO, el relé número 9 en la posición NC, y con eso, tendríamos conectado el cable Brown de los 220V del enchufe, y con el relé número 7 a NC y el número 10 a NO, tendríamos el cable Blue de la alimentación conectado.

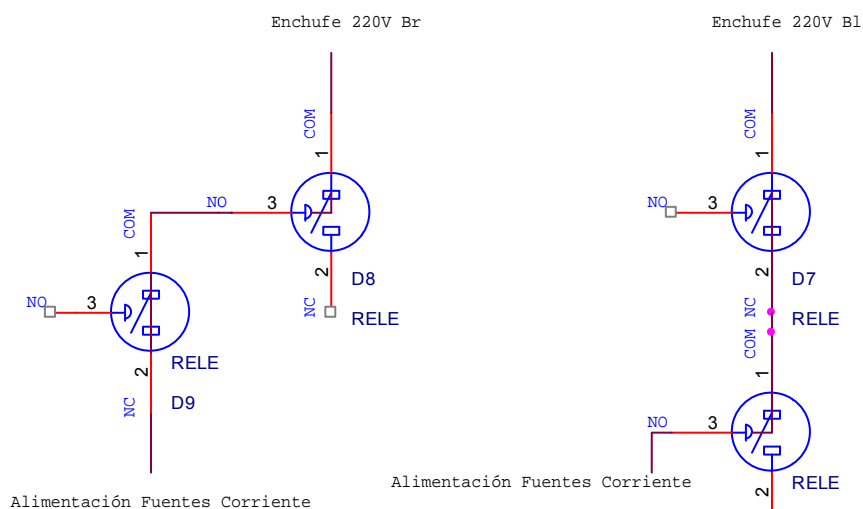


Figura 161 : Fuente corriente encendida

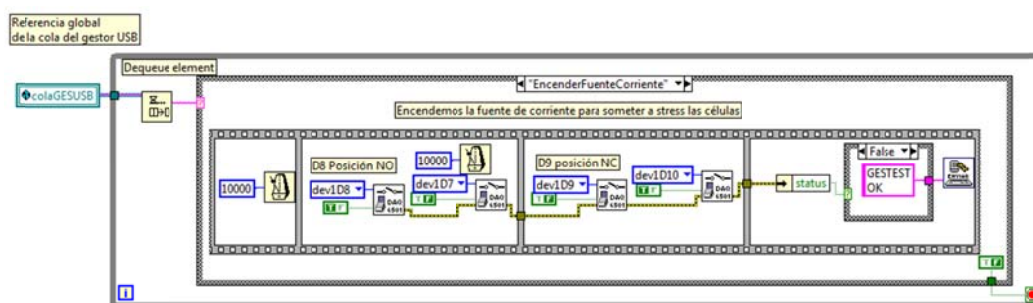


Figura 162: Encender Fuente Corriente

9.3.6. Salir

En esta opción se apaga la fuente de corriente al igual que en el estado anteriormente comentado, y se cierra el gestor USB.

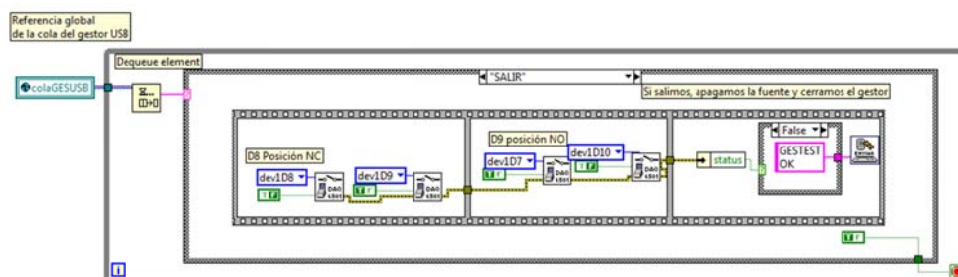


Figura 163: Salir.

9.4. Gestor Switcher

Este gestor es el encargado de comunicarse con el hardware switch Keithley 3706, el gestor se denomina, GESwitch_3706.vi, su desarrollo pertenece a otro proyecto (Belmonte), por lo que en este apartado se describirá ligeramente únicamente para comprender su funcionamiento y las menciones realizadas al mismo durante esta memoria..

El subvi contiene un switch con 12 estados

9.4.1. Inicializar

Permite abrir la sesión VISA con el dispositivo físico desde el PC. En nuestro caso, el GPIB era el 16.

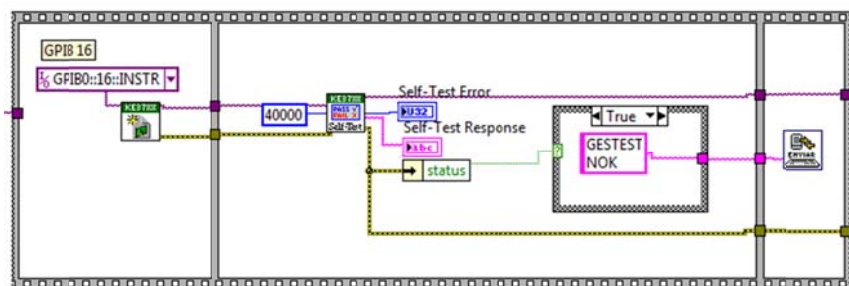


Figura 164: Inicializar

9.4.2. Abrir Canal

Permite abrir el canal seleccionado por el usuario, que se le pasan en la variable Lista de canales a abrir, junto con el canal 1911 o 1921 dependiendo del rango de canales en el que nos encontremos.

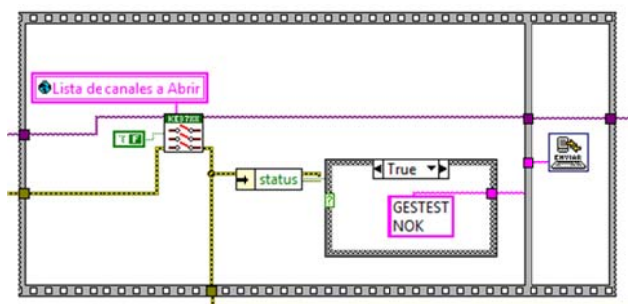


Figura 165: Abrir Canal

9.4.3. Abrir todos

Abre todos los canales del switcher, asegurando que ninguno queda cerrado.

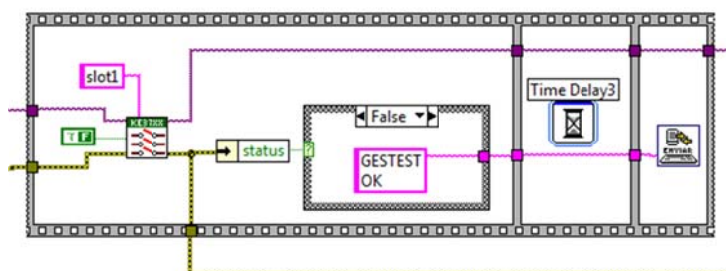


Figura 166: Abrir todos

9.4.4. Cerrar Canal

Permite cerrar los canales que se le pasan en la variable Switcher, la cual recibe el estado de los canales que desea cerrar el usuario de ControlEstadosSwitcher.vi y permite comprobar en una variable global que se han cerrado correctamente y sólo está cerrado el canal seleccionado por el usuario.

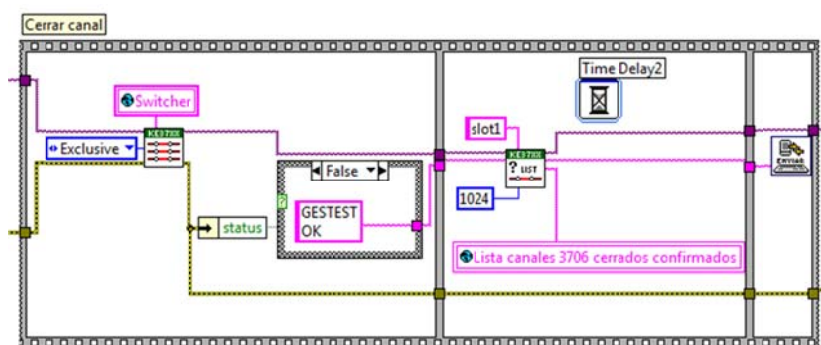


Figura 167: Cerrar Canal

9.4.5. Visualizar Cerrados

Comprobación vista en el punto anterior, en un estado independiente por si en un futuro se quisiera controlar por el usuario.

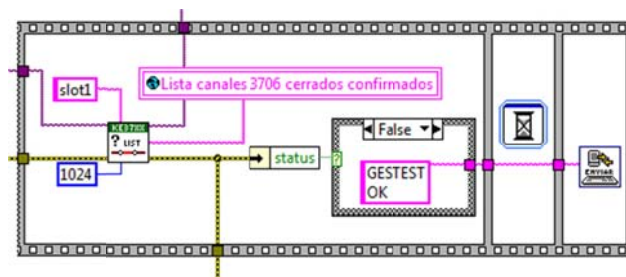


Figura 168: Visualizar Cerrados

9.4.6. Medir V

Permite medir la tensión que hay entre los dos terminales del canal del Keithley que previamente se debe haber cerrado. La medida se realizará en DC Volts y a 2 hilos. Posteriormente se almacena en la variable Lectura Medida V.

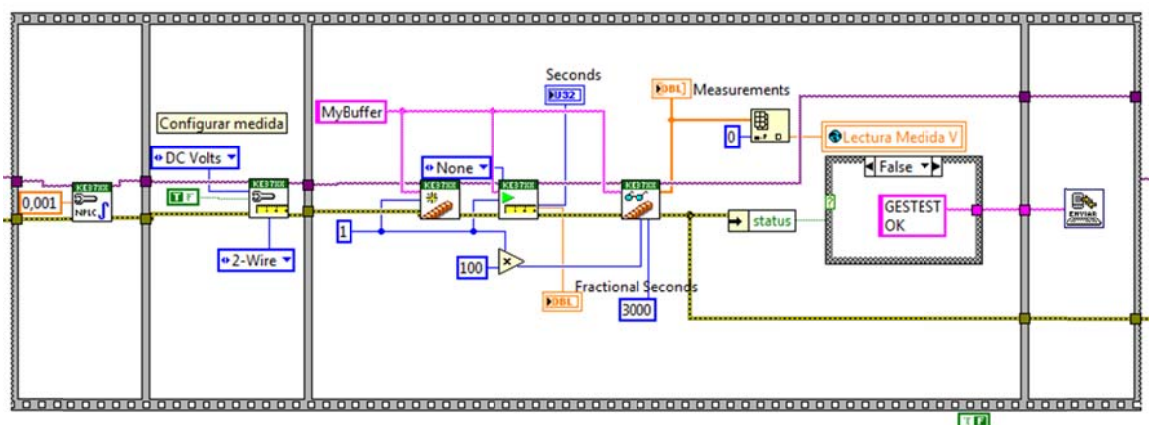


Figura 169: Medir V

9.4.7. Medir R

Permite medir la resistencia de un termoresistor colocado en un canal del Keithley, inicialmente el canal número 40, se mide resistencia a 2 hilos, se calcula la temperatura según la ecuación de la RTD $T_f = (R - 100) / 0.385$, siendo R el valor de resistencia medido y T_f la temperatura obtenida, y se almacena el valor en la variable global "Cadena Temperatura pt100".

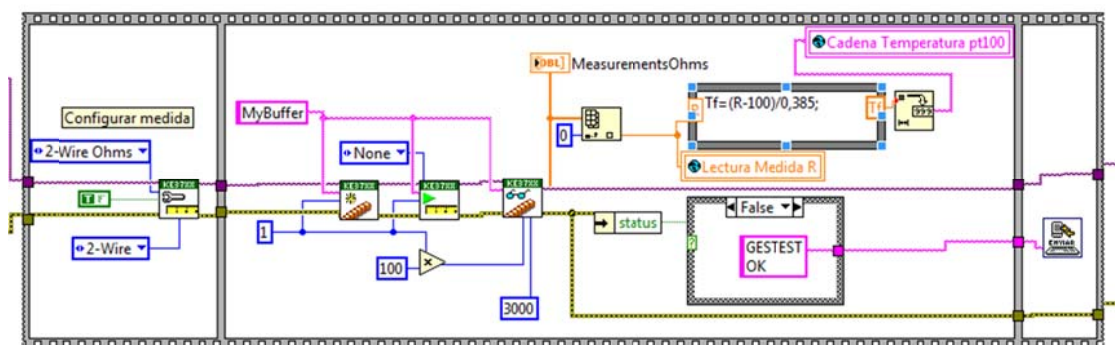


Figura 170: Medir R

9.4.8. Medir Termocouple

Permite medir la temperatura de un termopar conectado en un canal del Keithley, en este caso se mide temperatura a dos hilos, con un termopar de tipo K (K-Type), la referencia de temperatura es de 23° e interna. El valor de temperatura obtenido se almacena en la variable Cadena de Temperatura PT100.

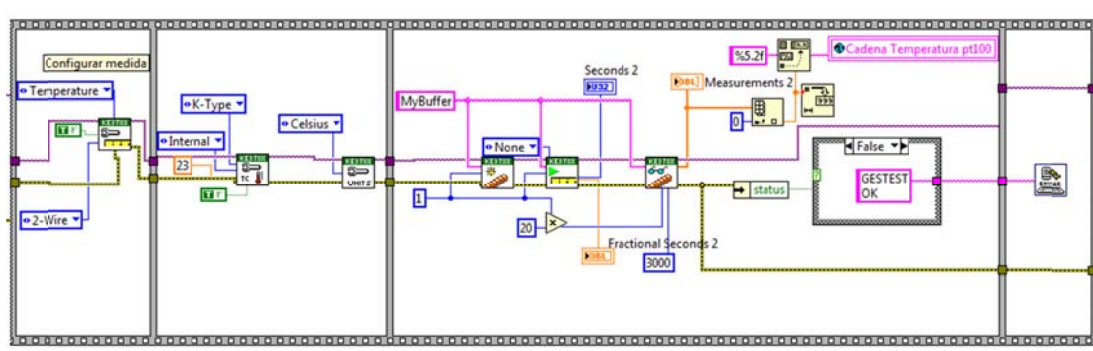


Figura 171: Medir Termocouple

9.4.9. Escribir Leer IO

Permite leer o escribir el canal digital de entrada salida que dispone el Keithley 3706, para ello debemos configurar la acción a realizar, lectura o escritura y se realiza de un bit o de un puerto.

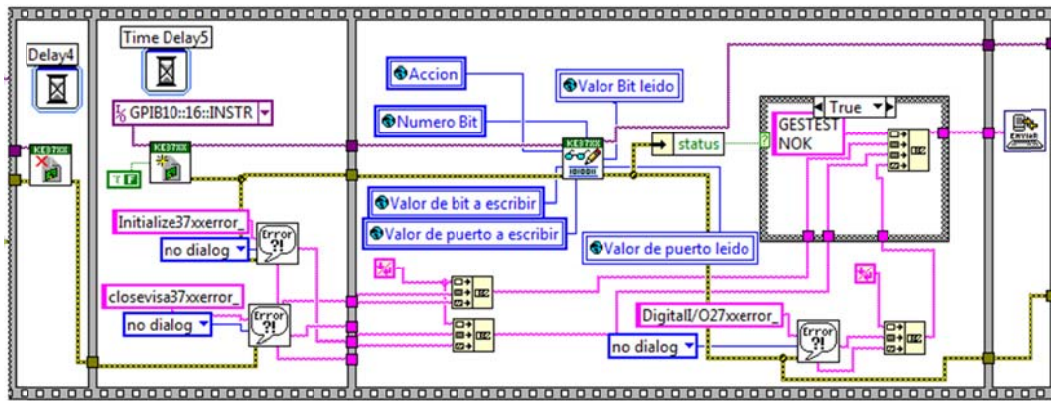


Figura 172: Escribir Leer IO

9.4.10. Cerrar

Permite medir cerrar la sesión VISA abierta con el Keithley, para finalizar la comunicación.

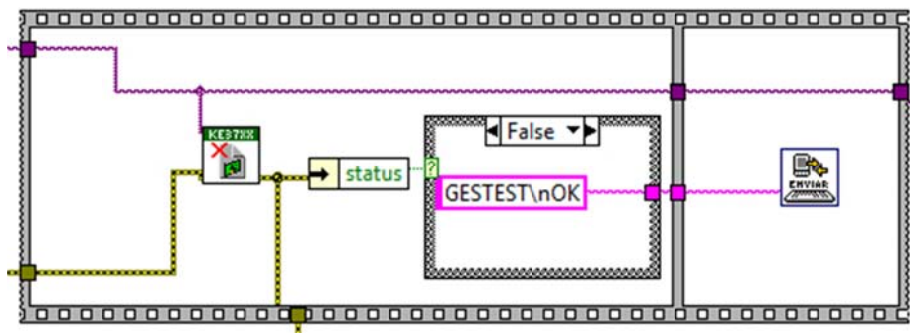


Figura 173: Cerrar

9.4.11. Salir

Al seleccionar este estado, indica que queremos cerrar la aplicación, por lo que para la ejecución del bucle while, cerrando el gestor del swithcer.

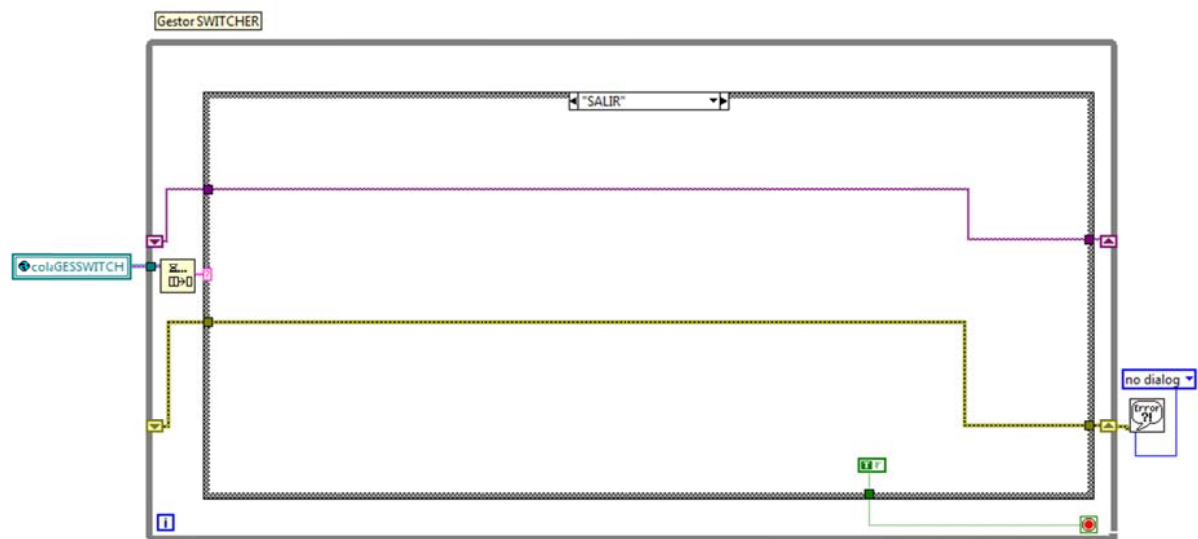


Figura 174: SALIR

9.5. Gestor alimentación.

Este gestor es el encargado de comunicarse con el hardware de la fuente de alimentación Agilent 6631B. Su desarrollo pertenece a otro proyecto (Belmonte), por lo que en este apartado se describirá ligeramente únicamente para comprender su funcionamiento y las menciones realizadas al mismo durante esta memoria.

9.5.1. Inicio

Permite abrir la sesión VISA con el dispositivo físico desde el PC. En este caso, el GPIB número 6, además configura los límites de protección de la fuente.

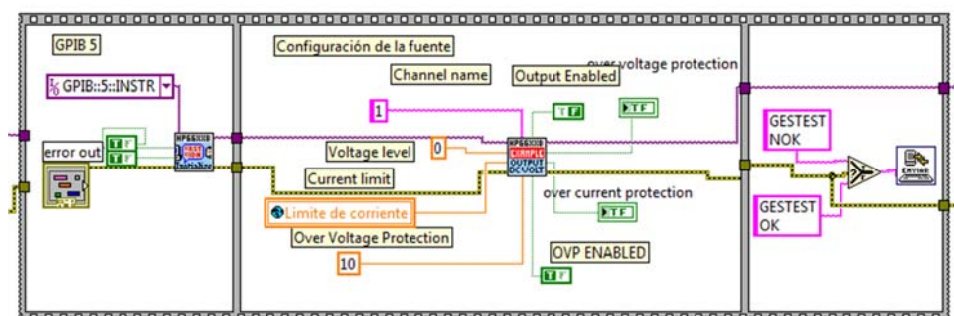


Figura 175: Inicio

9.5.2. Nivel Fuente

Establece la tensión seleccionada por el usuario en la fuente Agilent, y habilita la salida para poder caracterizar las células.

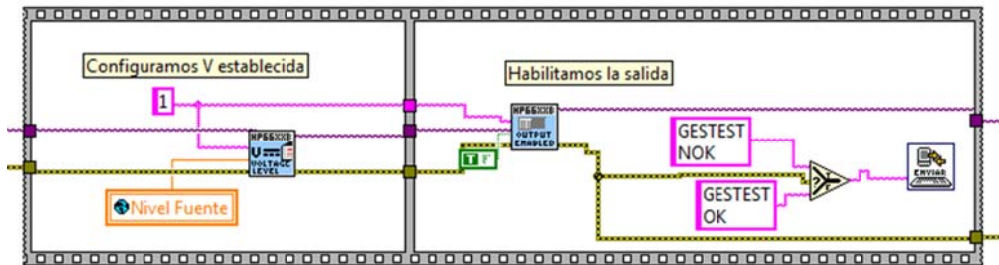


Figura 176: Nivel fuente

9.5.3. Medir

Configura el Agilent para medir la corriente que hay en la célula, toma la medida y la almacena en "lectura medida I".

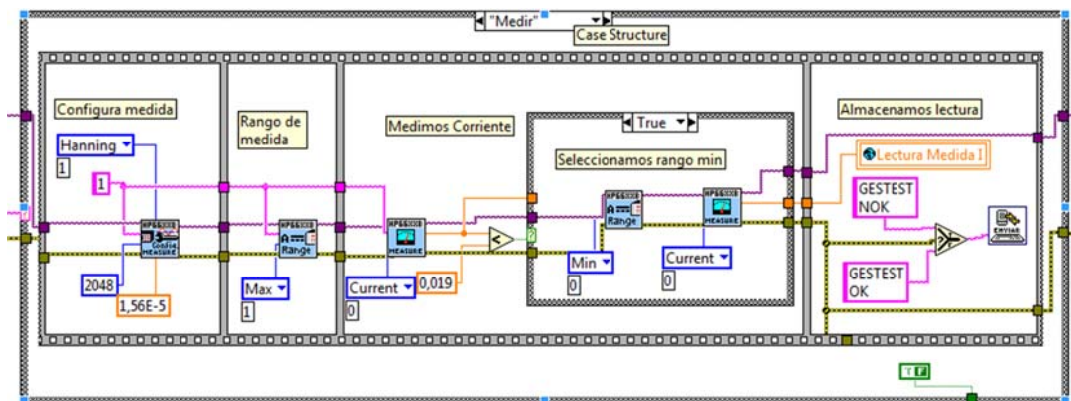


Figura 177: Medir

9.5.4. Cerrar

Establece un nivel de tensión de 0, y cierra la comunicación con el agilent.

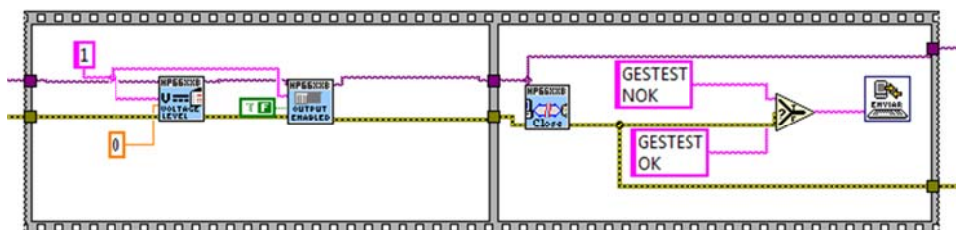


Figura 178: Cerrar

9.5.5. Salir

Al seleccionar este estado, indica que queremos cerrar la aplicación, por lo que para la ejecución del bucle “while”, cerrando el gestor del Agilent.

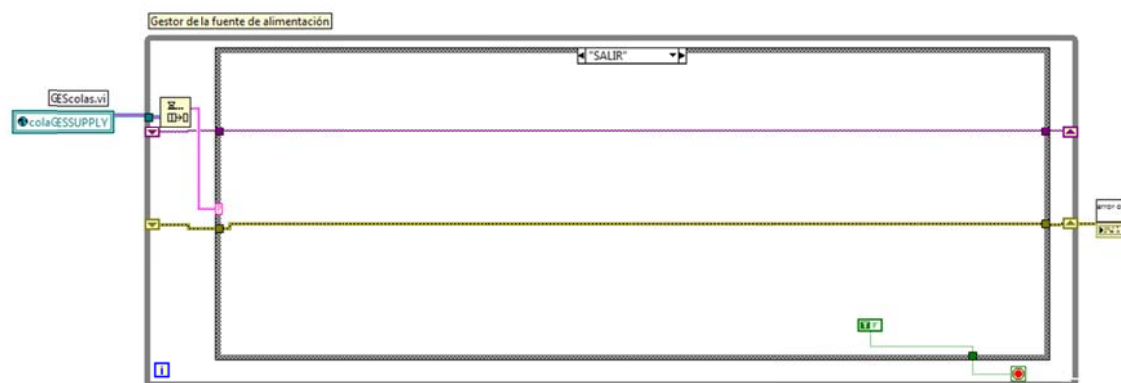


Figura 179: Salir

9.6. Controles definidos.

Para asegurar la correcta actualización de todas las copias del control de la máquina de estados, se han definido “enumerated type control”, creando un “type def enum”, de tal forma que si se añade o suprime algún estado, este se añade o suprime en todas las copias existentes. Esta opción es muy útil si se cometió algún error en el diseño de la máquina de estados, o si, como suele pasar, las especificaciones solicitadas en un primer borrador, no son las especificaciones necesarias del diseño final.

9.6.1. Control test2.ctf

El primer type def definido es el encargado del control de la máquina de estados definido en el gestest, el cual contiene los 18 estados definidos, de tal forma que para pasar de un estado a otro tan solo es necesario seleccionarle en el selector del enum, definido en estados TestMini.



Figura 180: Control test2.ctl

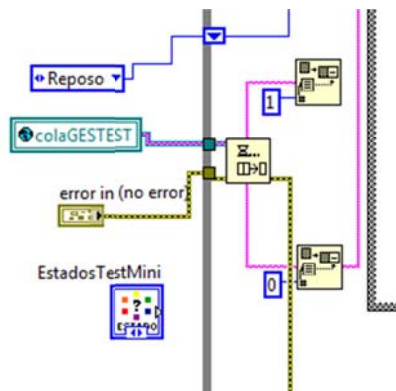


Figura 181: Definición del control

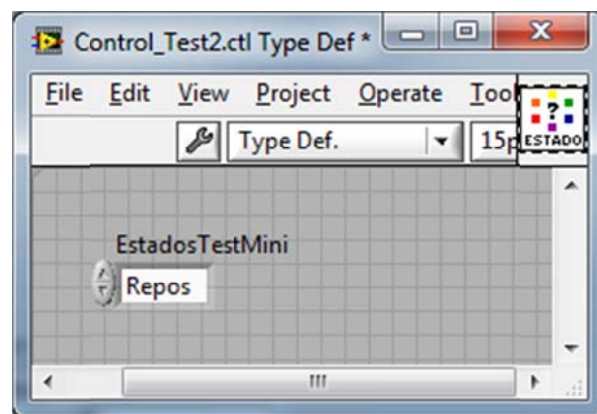


Figura 182: Type Def control estados

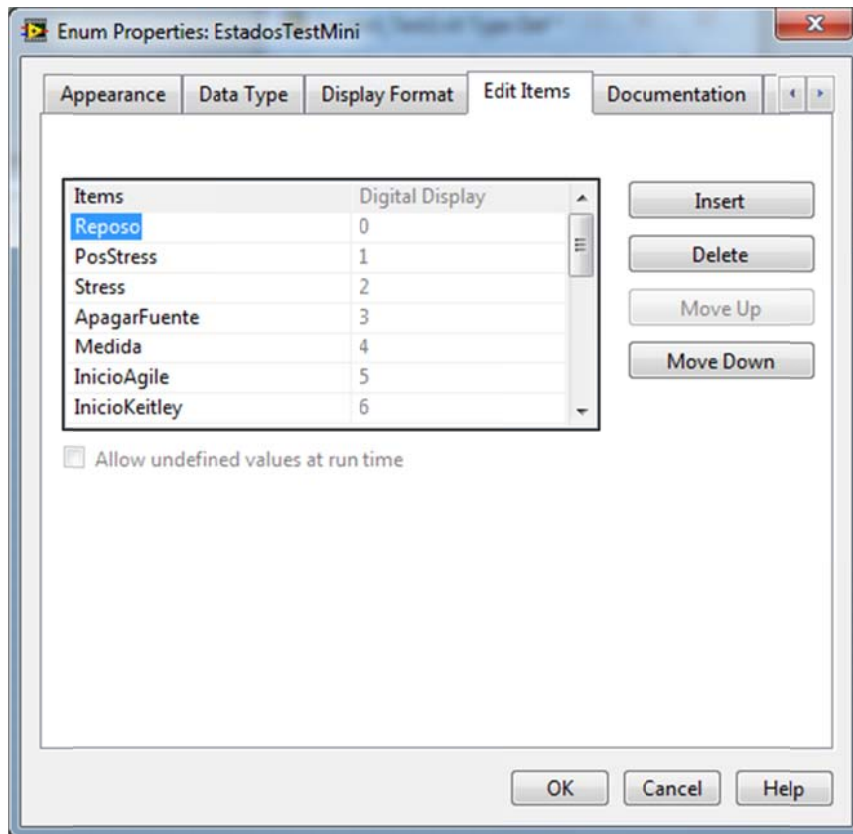


Figura 183: Definición de estados

9.6.2. Control switcher.ctl

Para la selección de los estados del control estados del switcher se ha creado un control del tipo enum descrito anteriormente, y se ha definido como type def para que todas sus copias se actualicen automáticamente al modificar un estado, en este estado, se han definido inicialmente los 40 canales correspondientes a las 40 células a medir, este número es debido a que la tarjeta 3721 del Keithley sólo dispone de 40 canales.

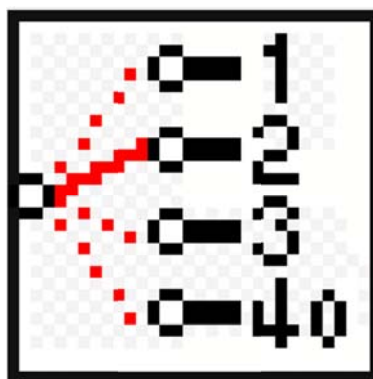


Figura 184: Control switcher.ctl

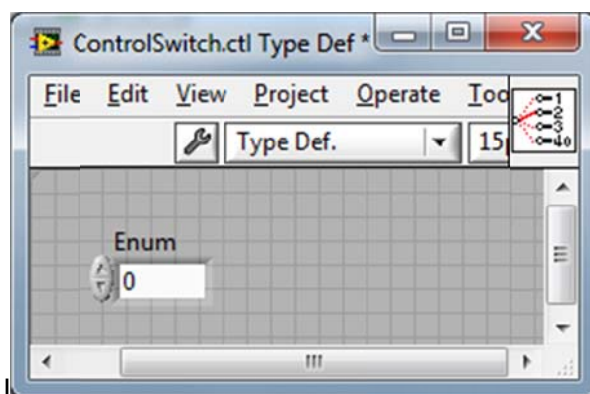


Figura 185: Type Def control switcher

9.6.3. DiodosMedir.ctl

Para la selección de los estados del control estados puertos, los correspondientes a los relés asociados a cada célula a medir, se ha creado un control del tipo enum descrito anteriormente, y se ha definido como type def para que todas sus copias se actualicen automáticamente al modificar un estado, en este estado, se han definido inicialmente los 40 canales correspondientes a los 40 relés de las células a medir, cada pareja de relés que controla la célula está asociado a un número de célula, distinguiendo en la llamada si es el relé correspondiente al puerto 2 o al puerto 0 asociado a dicha célula/diodo.

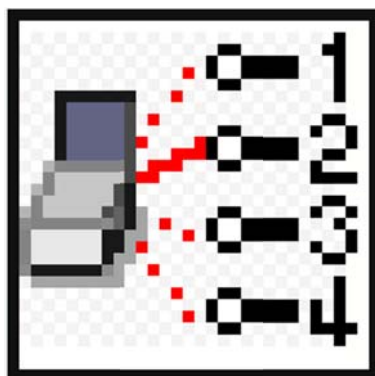


Figura 186: DiodosMedir.ctl

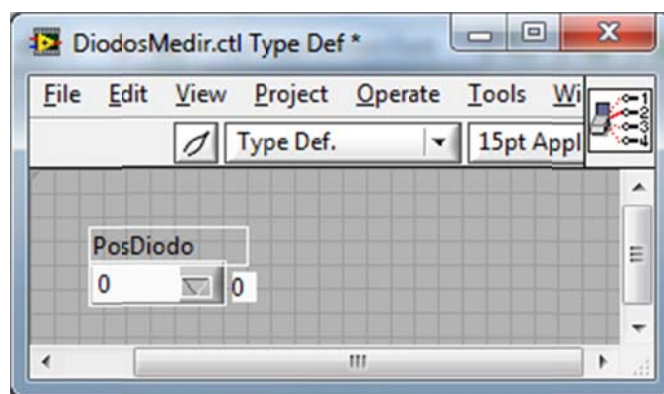


Figura 187: Type Def DiodosMedir.

9.7. Descripción VI desarrollados

9.7.1. ControlEstadosSwitcher.vi



Figura 188 : ControlEstadosSwitcher.vi

Contiene la asociación del número de célula con su correspondiente posición en el switcher Keithley, para ello está implementado con una estructura Case, de tal forma que dependiendo el estado seleccionado ejecutará un estado u otro, dentro de cada case tiene asociado la cadena de texto de los canales que deberá cerrar para realizar las medidas en la célula correspondiente.

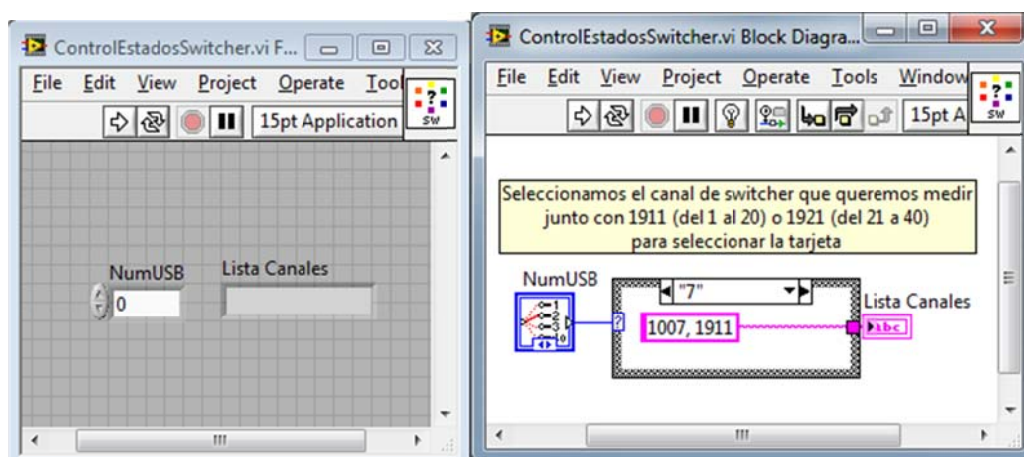


Figura 189 : Paneles Control Estados

Sigue la nomenclatura SCCC, donde S indica el número de tarjeta, y CCC el número de canal indicado con 3 dígitos, en la figura de arriba se puede observar que está en la tarjeta número 1, en la posición 007. Además para cerrar uno de los canales hay que indicarle además el número 1911 separado por coma, que indica que estamos usando la tarjeta 1 para medir a 2 hilos y coloca el canal seleccionado (sus dos hilos) a la entrada DMM del terminal, tal como se puede apreciar en la figura.

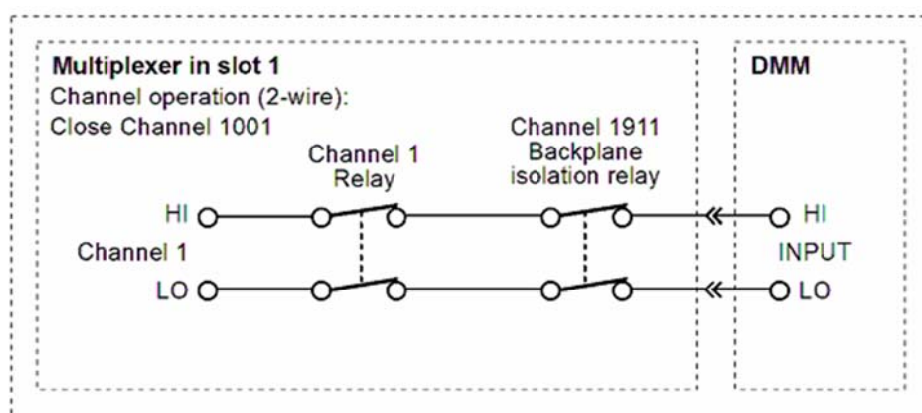


Figura 190 : Medida a 2 hilos

Si el canal seleccionado está entre el 1 y 20, es necesario cerrar el canal 1911, pero si está comprendido entre el 21 y el 40, hay que cerrar el canal 1922 para conectarle con el terminal DMM.

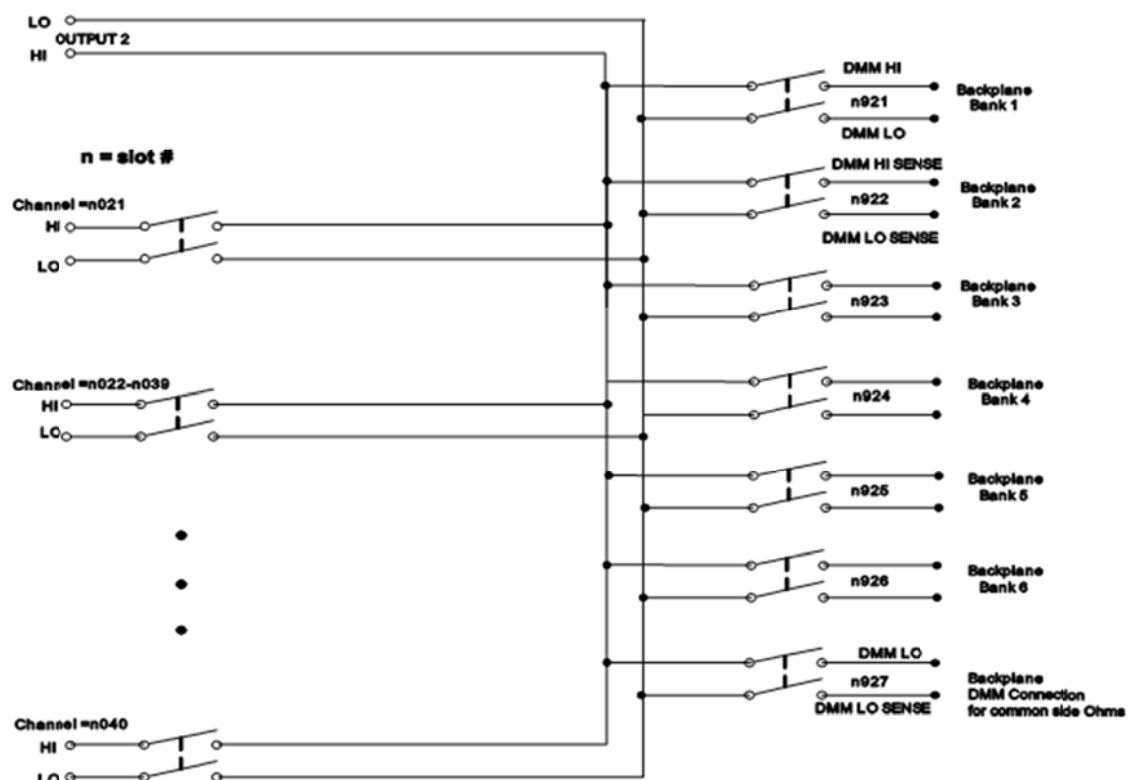


Figura 191: Interconexión tarjeta canales 21 al 40

Si se quisiera realizar la medida a 4 hilos sería necesario cerrar el segundo canal usado junto con el canal 1922, para conectar estos a la entrada SENSE y poder realizar las medidas.

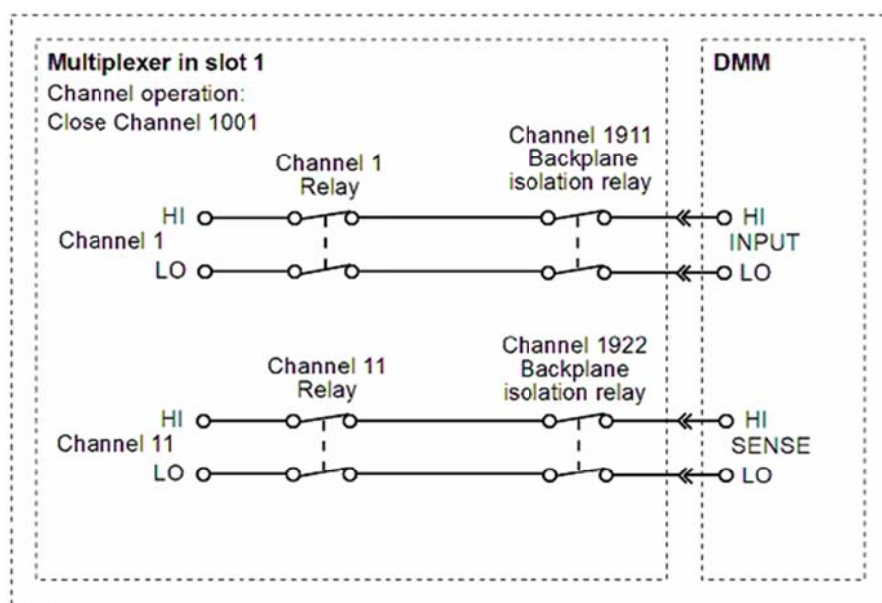


Figura 192 : Medida a 4 hilos

9.7.2. ControlEstadosPuertos.vi

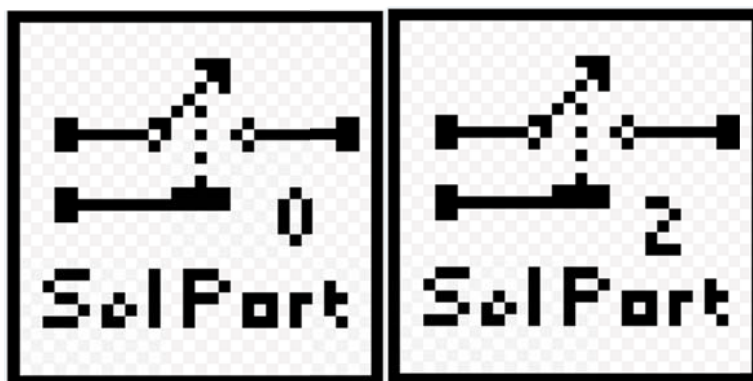


Figura 193 : ControlEstadosPuertos

A este programa se le pasa como parámetro el número de célula que queremos medir, y devuelve el código correspondiente para poder conmutar los relés mediante la señal digital del NI USB.

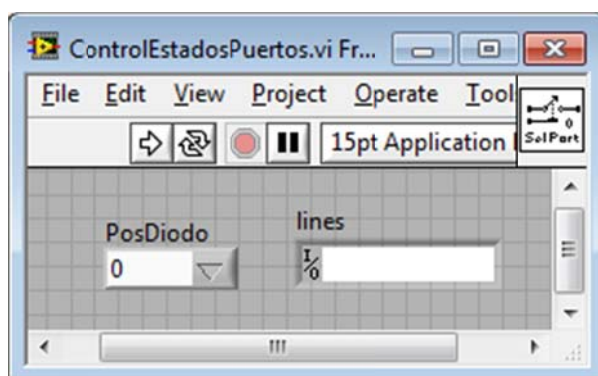


Figura 194: Panel frontal ControlEstadosPuertos.

En las siguientes figuras se aprecia que ambos programas son iguales, salvo en la salvedad que uno afecta al puerto 0 y el siguiente afecta al puerto 2, siendo los dev, los DAQ USB asociados a las tarjetas de relés correspondientes.

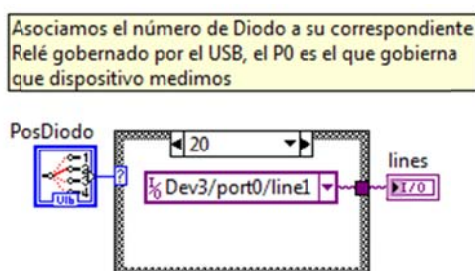


Figura 195: Estructura puerto 0

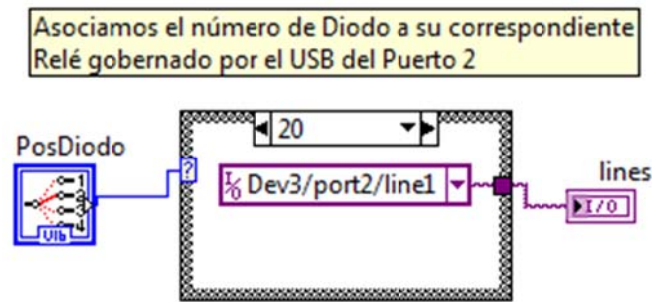


Figura 196: Estructura puerto 2

9.7.3. NumTarjetas.vi



Figura 197 : NumTarjetas.vi

Indica el número de tarjetas que se necesita tener conectadas en función de los canales que se desean medir, leerá como parámetros los canales que el usuario habrá seleccionado en el panel frontal, y en función de dichos valores, indicará al usuario el número de tarjetas que ha de tener conectadas para poder llevar a cabo los test y que el sistema no de error, mediante un led indicador del estado de cada tarjeta.

La primera tarjeta es necesaria para la alimentación, por lo que siempre deberá estar conectada.

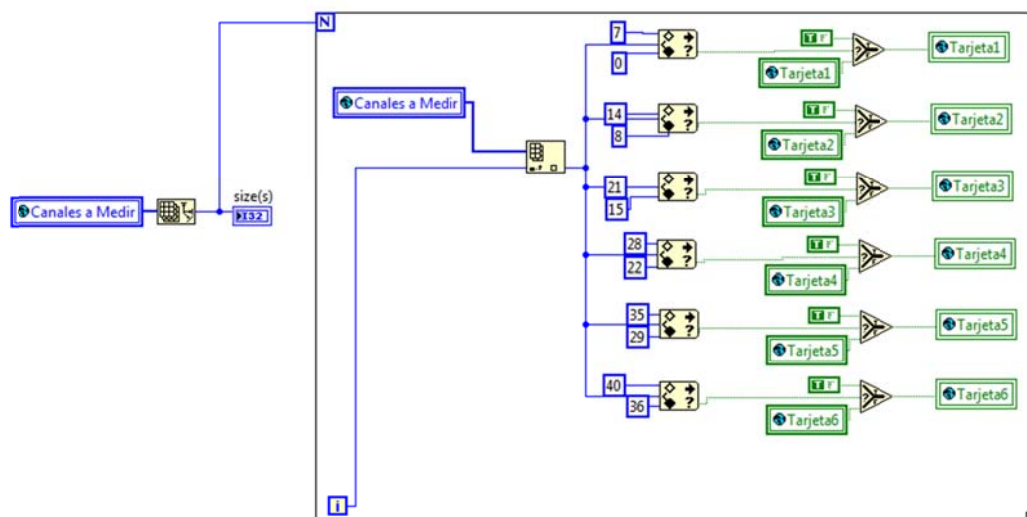


Figura 198: Conexión tarjetas.

9.7.4. Escribir_cabecera.vi



Figura 199 : escribir_cabecera.vi

En este módulo se le pasan los valores establecidos por el usuario denominados de cabecera, al módulo guardar cabecera, le pasa el número de célula que se está midiendo, el valor mínimo y máximo de alimentación seleccionado, la resolución de la fuente de alimentación, y el número de ronda de medida, así como la ruta de la curva.

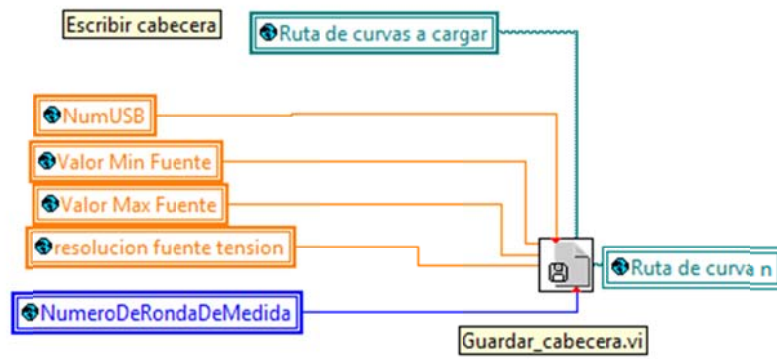


Figura 200 : escribir_cabecera

9.7.5. Guardar_cabecera.vi

Estructura para el almacenamiento de los datos resultantes de las operaciones realizadas, este subprograma contiene la estructura y variables de entrada y salida, con las que se le pasarán los datos que se han almacenado en las variables globales para su posterior tratamiento.

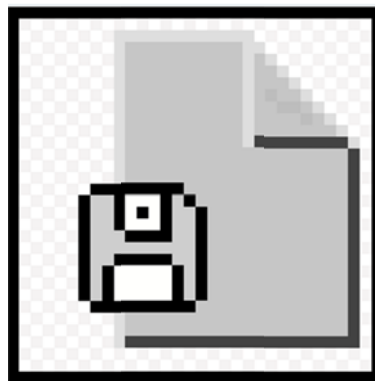


Figura 201 : guardar cabecera

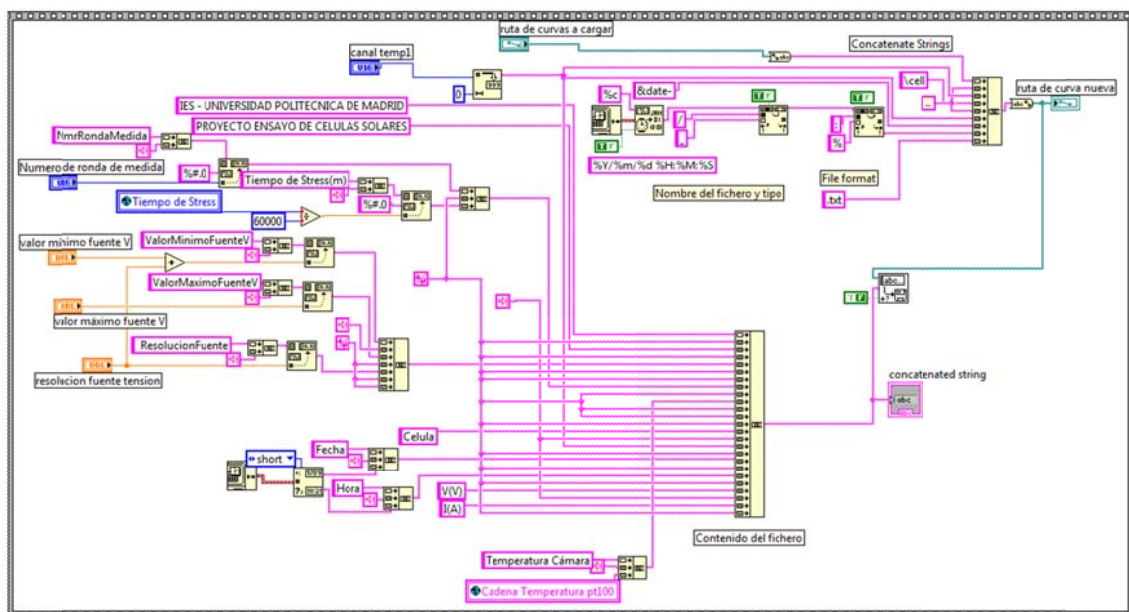


Figura 202 : Código guardar cabecera

Los datos del fichero se mostrarán de la siguiente forma:

IES - UNIVERSIDAD POLITECNICA DE MADRID

PROYECTO ENSAYO DE CELULAS SOLARES

ValorMinimoFuenteV	Vmax
ValorMaximoFuenteV	Vmin
ResolucionFuenteV	0,000000
NmrRondaMedida	N
Tiempo de Estrés	x minutos
Temperatura pt100	X °C
Celula	N
Fecha	07/02/2012
Hora	20:14
V(V)	I(A) (tabla de valores)

9.7.6. Usb_all.vi



Figura 203: Usb_all

En este VI se le pasa como parámetro el puerto de la tarjeta que se quiere control, y un booleano con la posición que debe tomar.

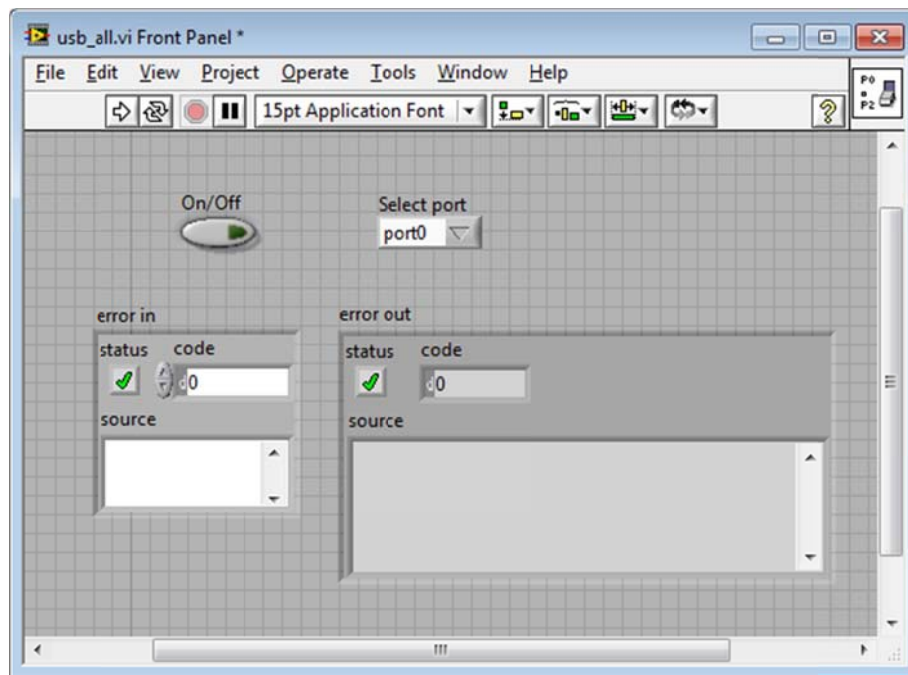


Figura 204: Panel frontal usb_all

El programa leerá de la variable global las tarjetas que el usuario debería tener seleccionadas, y aquellas que tuviera seleccionadas entrará dentro del primer case, una vez ahí, entrará dentro del segundo case, dependiendo si quiere acceder al puerto 0 o al puerto 2, genera la señal digital correspondiente a la tarjeta y puerto seleccionado, y escribe el valor booleano necesario para poner el puerto a NO o a NC según el caso.

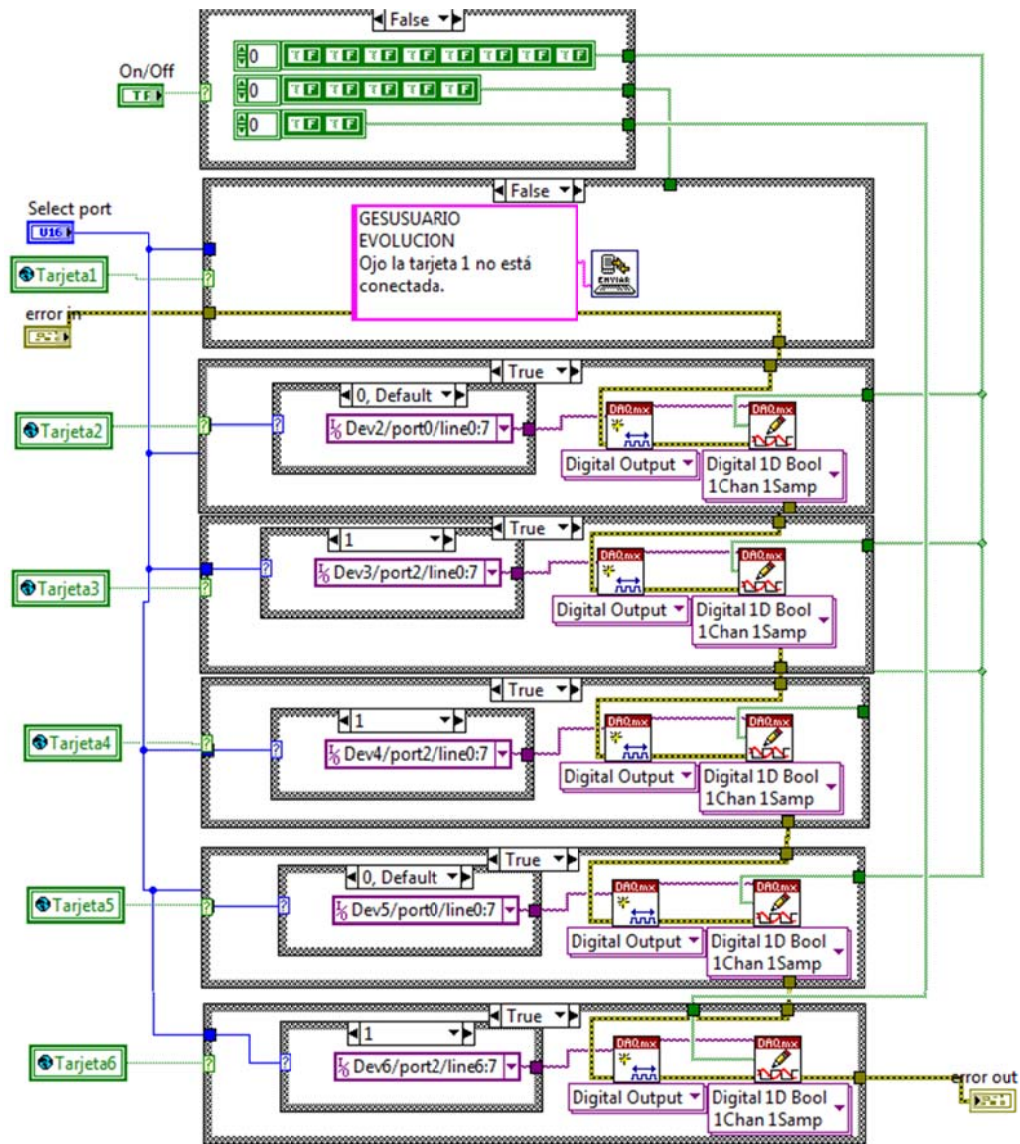


Figura 205: diagrama usb_all

9.7.7. USB_cell.vi



Figura 206: Usb_cell

Recibe como parámetro el código de canal necesario para acceder a NI USB y abrir o cerrar el relé correspondiente, dependiendo del dato booleano que se le haya introducido, TRUE o FALSE.

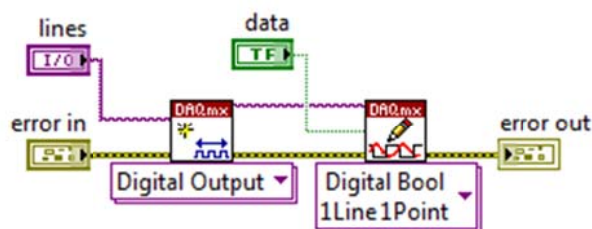


Figura 207: Código para abrir o cerrar un relé.

9.7.8. Usb.vi

Para la optimización de los recursos empleados, en este caso las tarjetas de relés, tras destinar 40 parejas de relés a la posible conexión de 40 células, y dejar en la primera tarjeta 3 parejas para el control de la alimentación 220V de las fuentes de corriente, en la sexta posible tarjeta sobrarían 3 parejas de relés más. Para optimizar su posible aprovechamiento o posibles mejoras, se ha optado por programar un subprograma en el que se pueda mediante un selector seleccionar cualquiera de los 12 relés mencionados de forma independiente, e indicar la posición que se desea obtener.



Figura 208: Usb.vi

En el panel frontal tenemos un selector que contiene el número de tarjeta etiquetado como dev1 o dev6, y el número de relé que se desee seleccionar. Además permite seleccionar el valor con TRUE/FALSE.

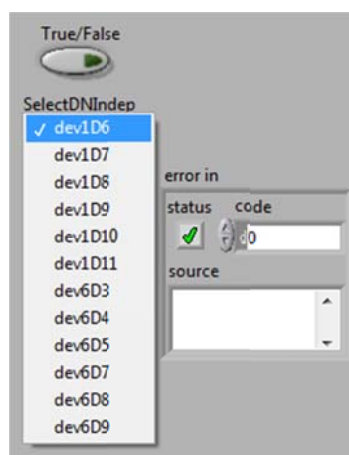


Figura 209: Panel frontal usb.vi

En el diagrama de bloques tendremos una estructura case, con el código correspondiente al relé seleccionado, tarjeta, puerto y línea, y a continuación se envía al DAQ la orden de poner a NO o NC el relé seleccionado.

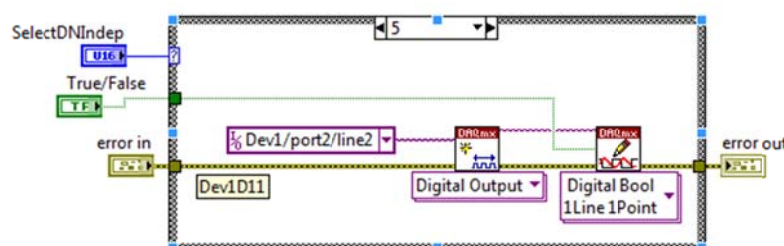


Figura 210: Diagrama de bloques usb.vi

9.7.9. ApagarFuentes.vi

Debido al número de veces que se usa la opción denominada apagar fuentes, para optimizar la programación, y debido a los problemas que ha habido para controlar los picos de corriente generados por las fuentes de corriente usadas para estresar las células, se ha optado por implementar este código en un subprograma aparte, de tal forma que si en un futuro se estropearan de nuevo los relés, tan solo habría que usar uno de los que actualmente quedan en reserva cambiándolo en el selector. Inicialmente se usaban dos parejas de relés para controlar la alimentación, como se ha comentado en la descripción hardware, dichos relés eran controlados por el programa, pero tras los problemas ocasionados por los picos de corriente que la tarjeta de relés no soportaba, se optó por programar con la misma filosofía de cruzamiento, una única pareja de relés, por la que solo pasan 5V para activar o desactivar una pareja de relés

externos que soportan la corriente generada por los generados de corriente al encenderse.



Figura 211: Apagar fuente

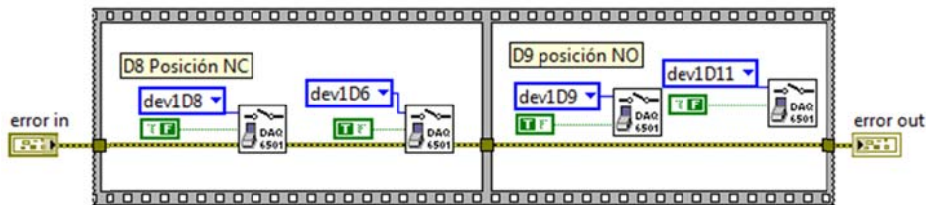


Figura 212 : Apagar Alimentación Fuente 4 relés con 220V

Como podemos observar en la figura, seleccionamos el relé independiente que queremos gestionar, y la posición que deseamos que este adopte si TRUE posición NO, o FALSE posición NC y llamamos a usb.vi.

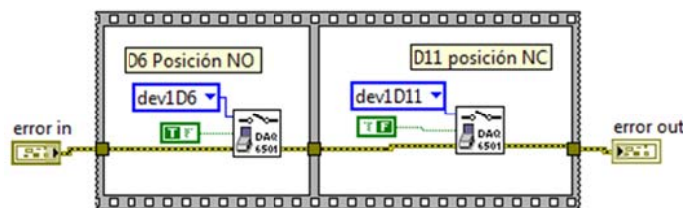


Figura 213: Apagar Alimentación Fuente 2Relés 5V

9.7.10. EncenderFuentes.vi

Inicialmente se usaban dos parejas de relés para controlar la alimentación, como se ha comentado en la descripción hardware, dichos relés eran controlados por el programa, pero tras los problemas ocasionados por los picos de corriente que la tarjeta de relés no soportaba, se optó por programar con la misma filosofía de

cruzamiento, una única pareja de relés, por la que solo pasan 5V para activar o desactivar una pareja de relés externos que soportan la corriente generada por los generados de corriente al encenderse.



Figura 214: Encender Fuentes

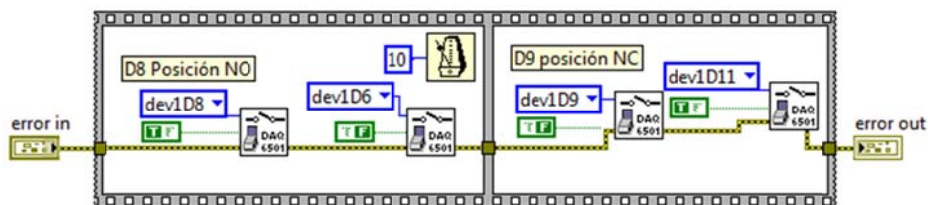


Figura 215: Encender Alimentación Fuentes con 4 relés a 220V

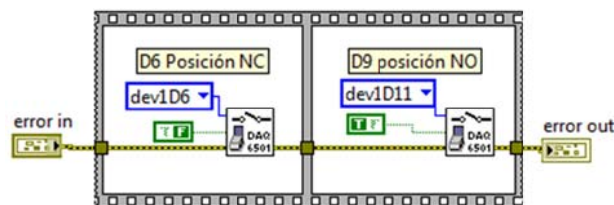


Figura 216: Encender Fuentes con 2 relés a 5V

9.7.11. minMax.vi



Figura 217: min Max.vi

Se establece el valor de tensión que ha de ponerse en la fuente de alimentación Agilent 6631B, dicho valor se obtiene de los parámetros introducidos por el usuario, según la ecuación:

$$V = \left\{ \left[\frac{(V_{Max} - V_{min})}{\text{NumeroMedidas}} \right] \times \text{NumRonda} \right\} + V_{min} + IR$$

Dónde Vmax es la tensión máxima, Vmin la tensión mínima, Número de medidas el número de muestras total que desea tomar el usuario, y NumRonda, el número de la medida en la que se encuentra actualmente.

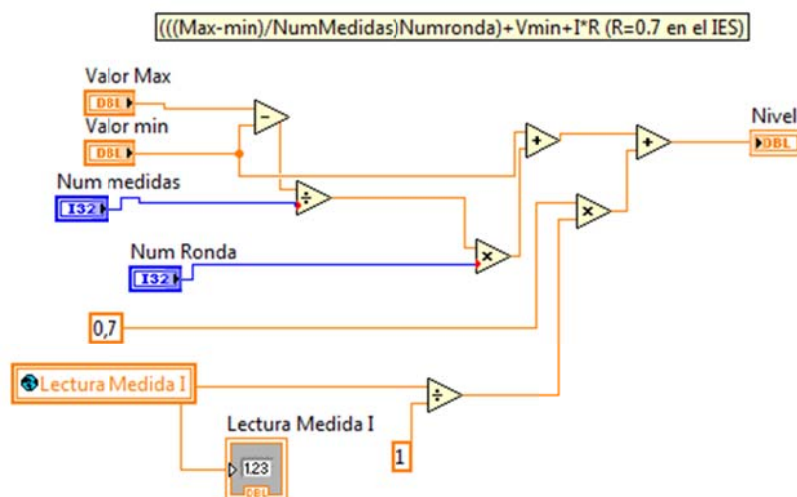


Figura 218: Ecuación minMAX

9.7.12. Maxmin.vi

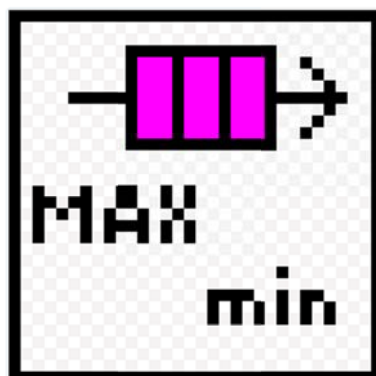


Figura 219:MAXmin.vi

Debido al calentamiento de la célula al realizar la medida, se optó por dejar a elección del IES el modo de hacer la toma de medidas, si de menor a mayor, provocando un leve calentamiento limitando levemente la corriente final obtenida como se verá en las gráficas en el apartado de puesta en marcha, o realizar la medida de mayor a menor, viendo una ligera variación en la curva obtenida. Con el primero de los casos el resultado obtenido es exacto a la caracterización de la célula.

En la siguiente figura se muestra el código programado para realizar la medida de mayor a menor, siguiendo la siguiente ecuación:

$$V = V_{max} - \left\{ \left[\frac{(V_{Max} - V_{min})}{\text{NumeroMedidas}} \right] \times \text{NumRonda} \right\} - IR$$

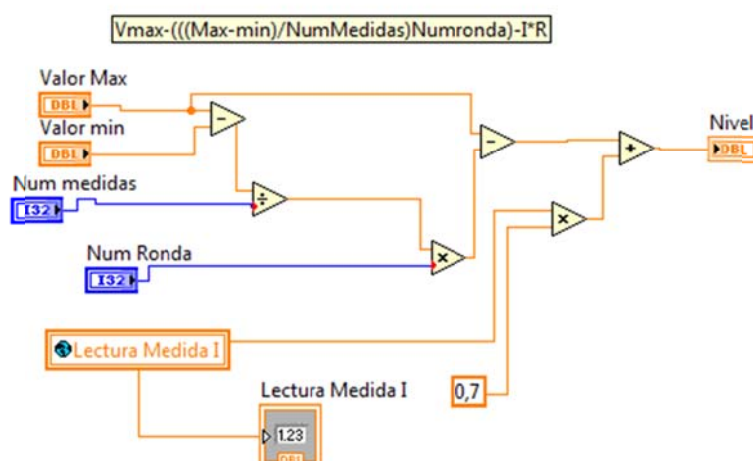


Figura 220: Ecuación MAXmin

Se ha configurado un panel frontal exacto con las mismas entradas y salidas para ambos subprogramas, de tal forma que únicamente hay que dar al botón reemplazar uno por otro, y el programa seguiría funcionando a la perfección con la salvedad mencionada.

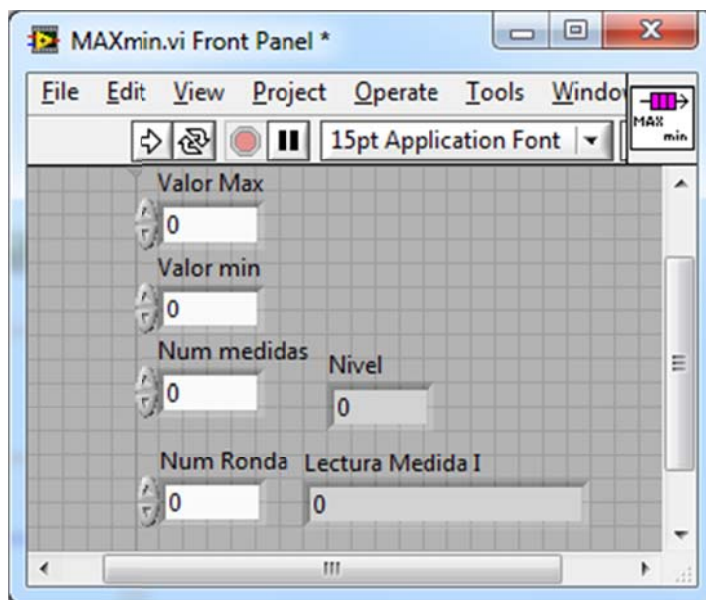


Figura 221: Panel frontal MAXmin

9.8. Variables globales

En este apartado se encuentran todas las variables globales usadas y mencionadas a lo largo de este proyecto, son útiles para la verificación del correcto funcionamiento del programa durante su fase de desarrollo, y puesto que este proyecto está destinado no solo a los dos proyectos del IES, uno en el propio IES que se ha dejado instalado como se verá durante este proyecto, y otro que se comenzará en breve en las instalaciones de la EUIT de Telecomunicación como parte de una tesis, sino también a su posible reutilización para ensayos con otro tipo de células u otros dispositivos que requieran la misma filosofía de funcionamiento a la hora de realizar ensayos acelerados, para los cuales quizá sea necesario hacer alguna ligera modificación en el código.

Las variables globales están divididas en dos paneles, el primero de ellos relativo al control de usuario.

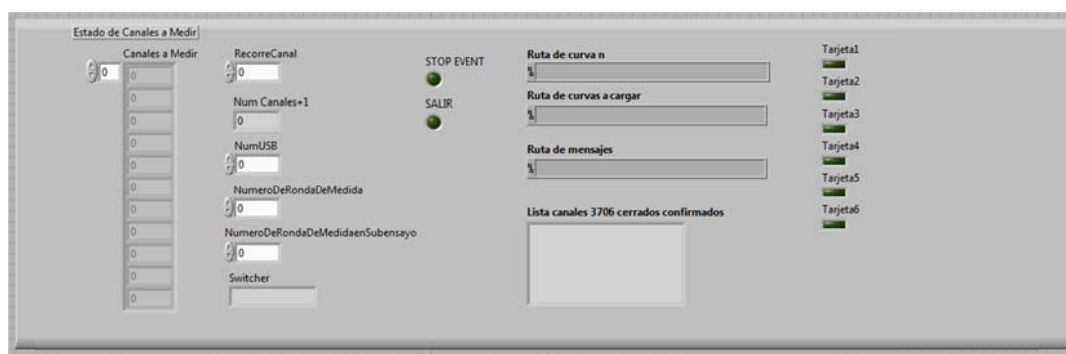


Figura 222: Variables globales usuario

En este panel tenemos la lista de canales que el usuario ha seleccionado para tomar medidas.

Recorre canal, variable usada para recorrer la lista de canales a medir.

Numcanales+1 que indica el número de canales seleccionado por el usuario más 1, puesto que al crear la tabla se inicia con un primer valor nulo.

NumUSB indica la célula seleccionada que se desea medir.

NumeroDeRondaDeMedida indica el número de medidas realizadas al conjunto de todas las células seleccionadas, dicho de otro modo, el número de ciclos estrés/medida realizados.

NumeroDeRondaDeMedidaSubensayo es la variable de control para llevar la cuenta de las medidas que se han realizado hasta el momento, y se realizarán hasta que este valor sea igual al NumeroDeRondaDeMedida.

Switcher es la variable que contiene una cadena de texto con los valores necesarios para cerrar el canal en el switcher keithley

STOP Event, led indicador que se ha pulsado el botón de Stop.

Salir, led indicador de que se ha pulsado el botón salir.

Ruta de la curva n: Ruta del archivo de cada medida de cada célula.

Ruta de curvas a cargar: Ruta general para encontrar los ficheros previamente creados.

Ruta de mensajes: Ruta del fichero que contiene todos los mensajes enviados en la sesión a través de GESmensajes.vi.

Lista de los canales del 3706 cerrados, variable útil para la verificación del canal actual que está cerrado en el keithley y que está siendo medido, es muy útil su verificación en este apartado ya no solo para un posible control en remoto sino porque el display del switcher keithley no indica el canal que está cerrado en ese momento, únicamente indica el 1001 siempre.

Tarjeta N, hay un led por cada tarjeta que se puede conectar en el sistema, el cual, según los canales seleccionados a medir, indica el número de tarjetas que tenemos que tener instaladas y funcionando, sino lo estuvieran el sistema dará error.

El siguiente panel de variables globales incluye la configuración de las medidas, tanto lo relacionado con los tiempos de estrés como con la configuración del Agilent y del Keithley.

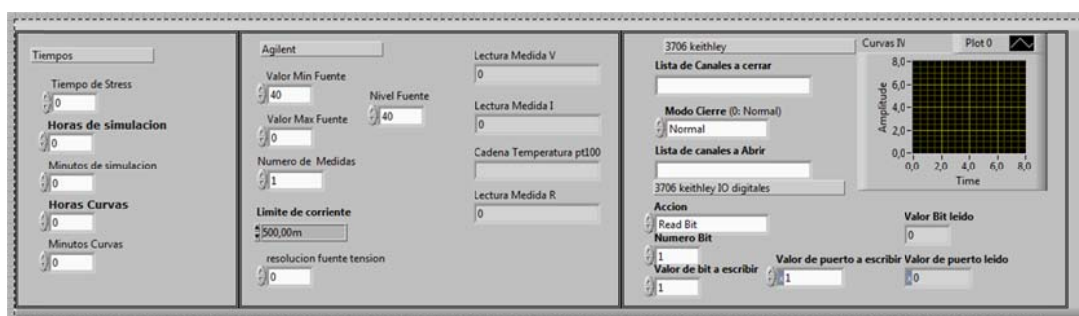


Figura 223: Variables globales medidas

En el apartado Tiempos tenemos todo lo relacionado con los tiempos de estrés y de reposo.

Tiempo de stress, variable que contiene el tiempo acumulado de estrés al que han estado sometidas las células hasta el momento.

Horas y minutos simulación, variable configurada por el usuario que contiene el tiempo de estrés que desea que tengan las células en cada ciclo.

Horas y minutos curvas, variable configurable por el usuario que contiene el tiempo de reposo que se le deja a las células una vez finalizado el tiempo de estrés y antes de empezar a tomar medidas.

Tiempos

Tiempo de Stress
0

Horas de simulacion
0

Minutos de simulacion
0

Horas Curvas
0

Minutos Curvas
0

Figura 224: Variables globales, tiempos

En el siguiente panel, está la configuración del Agilent, por parte del usuario, en el cual indica el valor mínimo y máximo de la fuente, entre los que desea tomar medidas, el número de medidas a tomar entre dichos valores, así como una limitación de corriente como medida de protección de la célula. Además se indica en la variable nivel fuente, el valor establecido por la fuente de alimentación Agilent en cada momento.

Agilent

Valor Min Fuente
40

Valor Max Fuente
0

Numero de Medidas
1

Limite de corriente
500,00m

Nivel Fuente
40

Figura 225: Variables globales Agilent

En el siguiente panel están las variables globales relacionadas con las medidas, en la que se encuentran las medidas de la curva IV, la medida de la resistencia si lo que hemos usado para medir la temperatura es una pt100, y la temperatura leída, independientemente de que se haya usado la pt100 o el termopar.

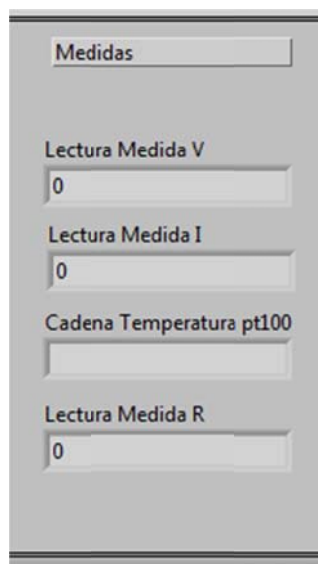


Figura 226: Variables globales, medida.

Y por último las variables relacionadas con el keithley, la lista de canales a abrir y a cerrar y el modo en el que se cierra, así como una gráfica en la que se mostraban las curvas IV en la fase de desarrollo, para comprobar los datos almacenados.

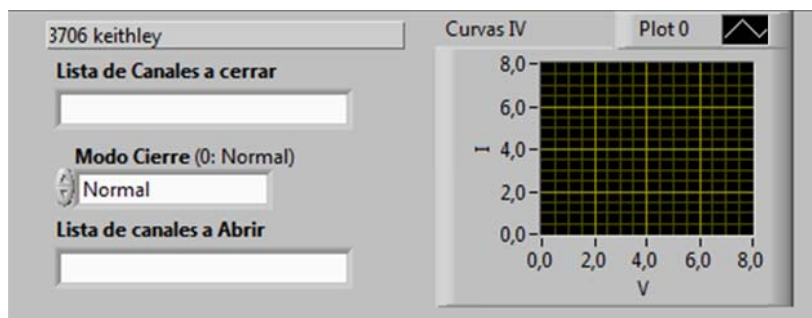


Figura 227: Variables globales, keithley

10 Ensayos acelerados. Puesta en marcha.

10.1. Primeras pruebas

Para comprobar el correcto funcionamiento del software y del diseño hardware, se hizo un primer montaje de conexionado usando diodos para verificar el correcto funcionamiento de todos los componentes.

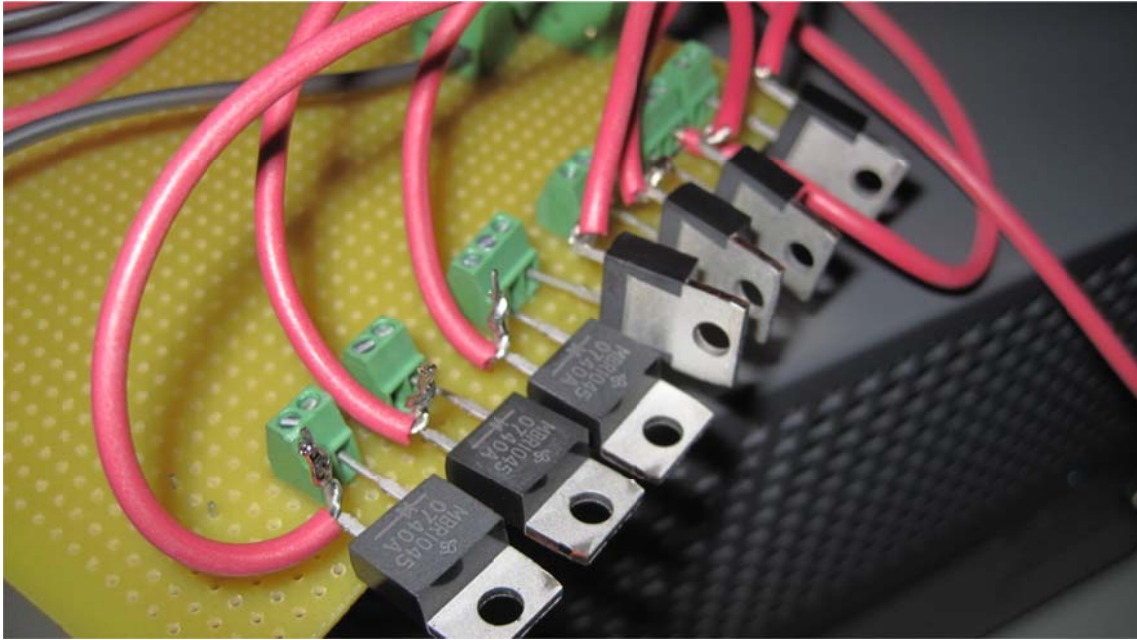


Figura 228: Diodos testeados

Para facilitar el testeo de todos los componentes se optó por montar el siguiente esquema de interconexiones en una placa aparte.

KEIT 1H	RELÉ 1 COM	DIODO 1 +
KEIT 1L	RELÉ 16 COM	DIODO 1 -
FUENTE 1 +	RELÉ 1 NO	
FUENTE 1 -	RELÉ 16 NO	
KEIT 2H	RELÉ 2 COM	DIODO 2 +
KEIT 2L	RELÉ 15 COM	DIODO 2 -
FUENTE 2 +	RELÉ 2 NO	
FUENTE 2 -	RELÉ 15 NO	
KEIT 3H	RELÉ 3 COM	DIODO 3 +
KEIT 3L	RELÉ 14 COM	DIODO 3 -
FUENTE 3 +	RELÉ 3 NO	
FUENTE 3 -	RELÉ 14 NO	
KEIT 4H	RELÉ 4 COM	DIODO 4 +
KEIT 4L	RELÉ 13 COM	DIODO 4 -
FUENTE 4 +	RELÉ 4 NO	
FUENTE 4 -	RELÉ 13 NO	
KEIT 5H	RELÉ 5 COM	DIODO 5 +
KEIT 5L	RELÉ 12 COM	DIODO 5 -
FUENTE 5 +	RELÉ 5 NO	
FUENTE 5 -	RELÉ 12 NO	
KEIT 6H	RELÉ 6 COM	DIODO 6 +
KEIT 6L	RELÉ 11 COM	DIODO 6 -
FUENTE 6 +	RELÉ 6 NO	
FUENTE 6 -	RELÉ 11 NO	
Alimentacion	RELÉ 7 COM	
RELE 7 NC	RELÉ 10 COM	
RELÉ 10 NO	Alimentación fuentes	

Agilent +	RELÉ 1 NC	RELÉ 2 NC	RELÉ 3 NC	RELÉ 4 NC	RELÉ 5 NC	RELÉ 6 NC
Agilent -	RELÉ 16 NC	RELÉ 15 NC	RELÉ 14 NC	RELÉ 13 NC	RELÉ 12 NC	RELÉ 11 NC

Tabla VI: Conexionado tarjeta pruebas

Una vez hecho el esquema se pasa a cablear la tarjeta con la tarjeta de 16 relés, y esta a su vez con la tarjeta del keithley para verificar la correcta conmutación de los relés y verificar que el recorrido de los canales por parte del software es correcto.

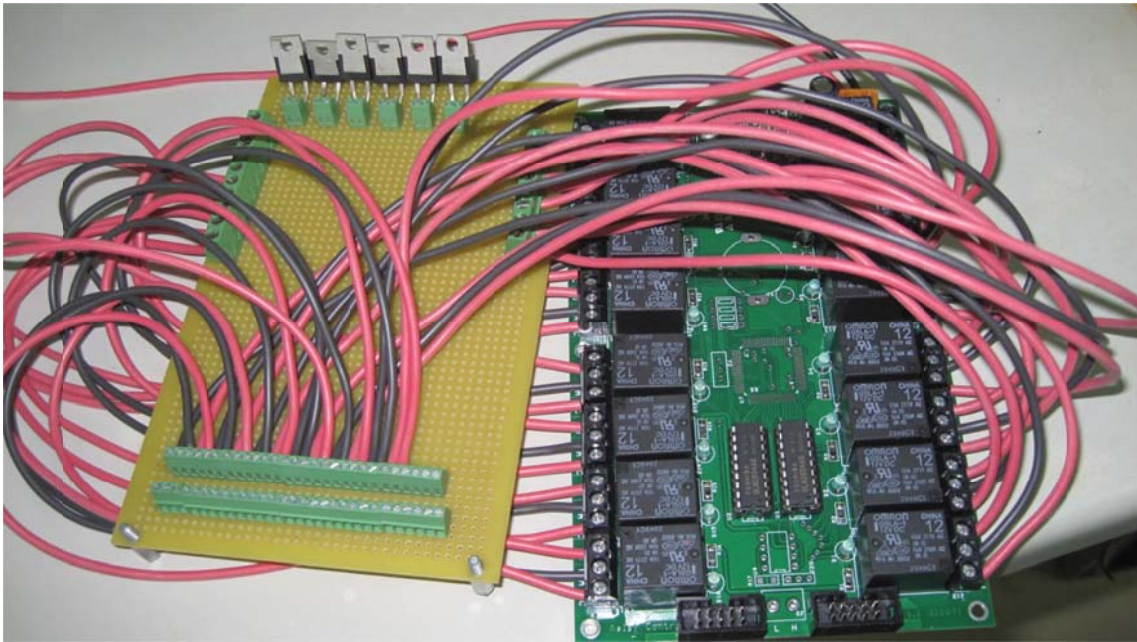


Figura 229: Interconexión tarjeta-diodos

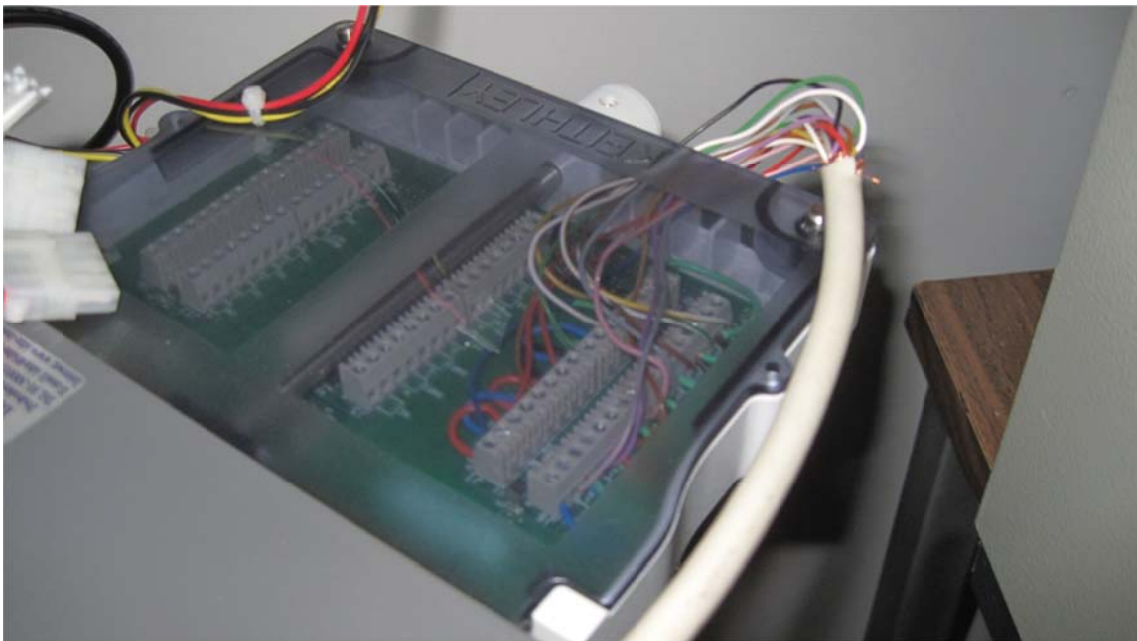


Figura 230: Primer montaje tarjeta 3721

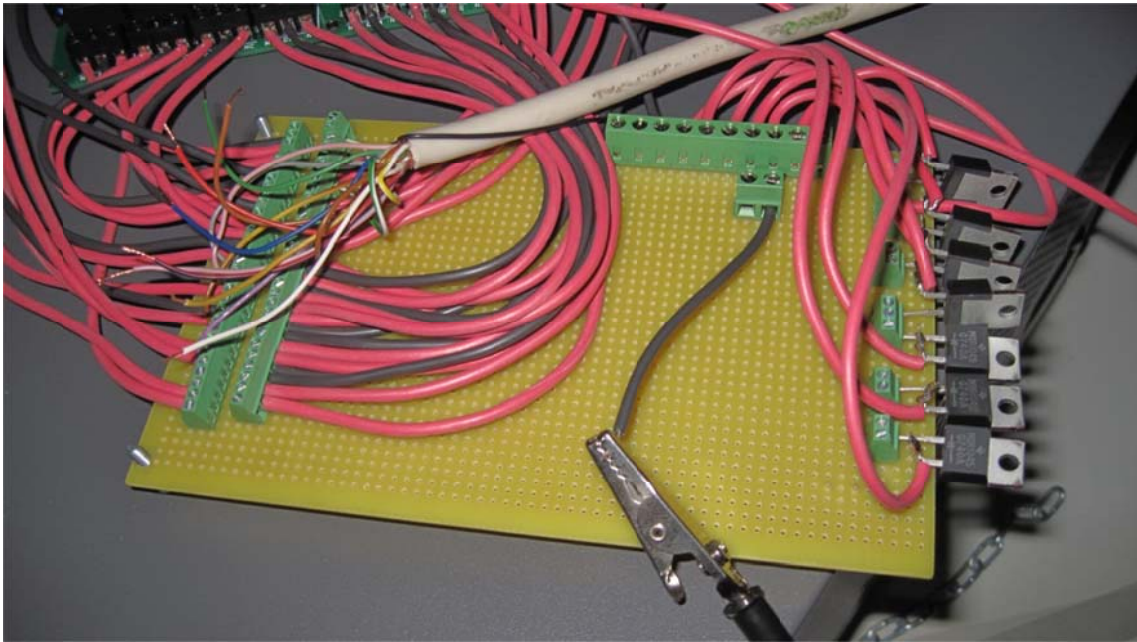


Figura 231: Interconexión Tarjeta 3721 con diodos

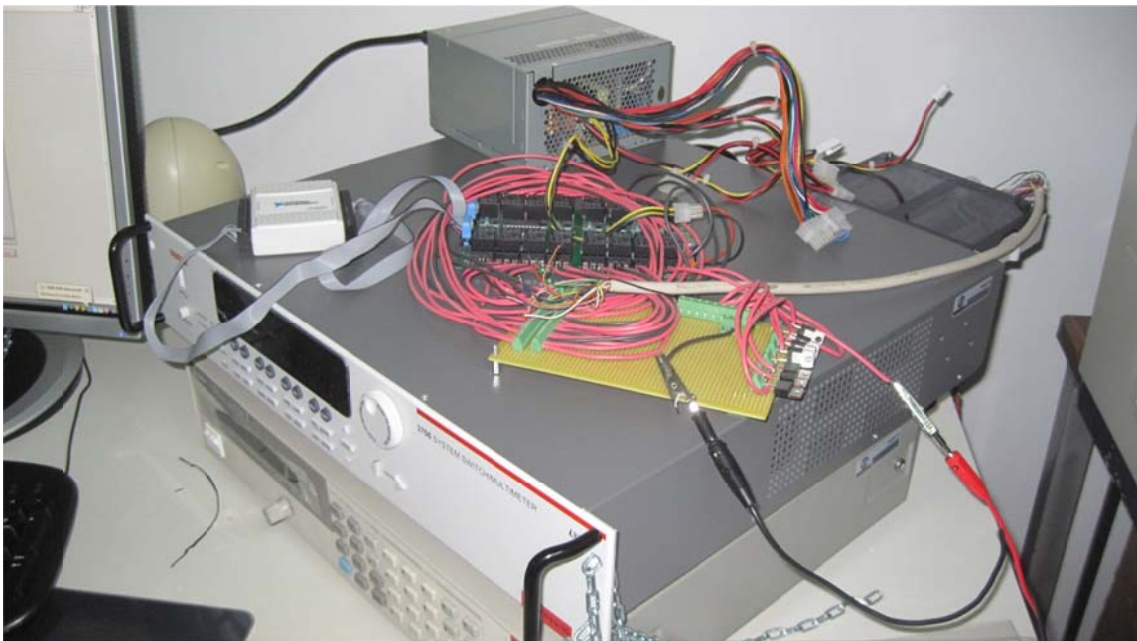


Figura 232 : Primer montaje con 1 tarjeta de relés

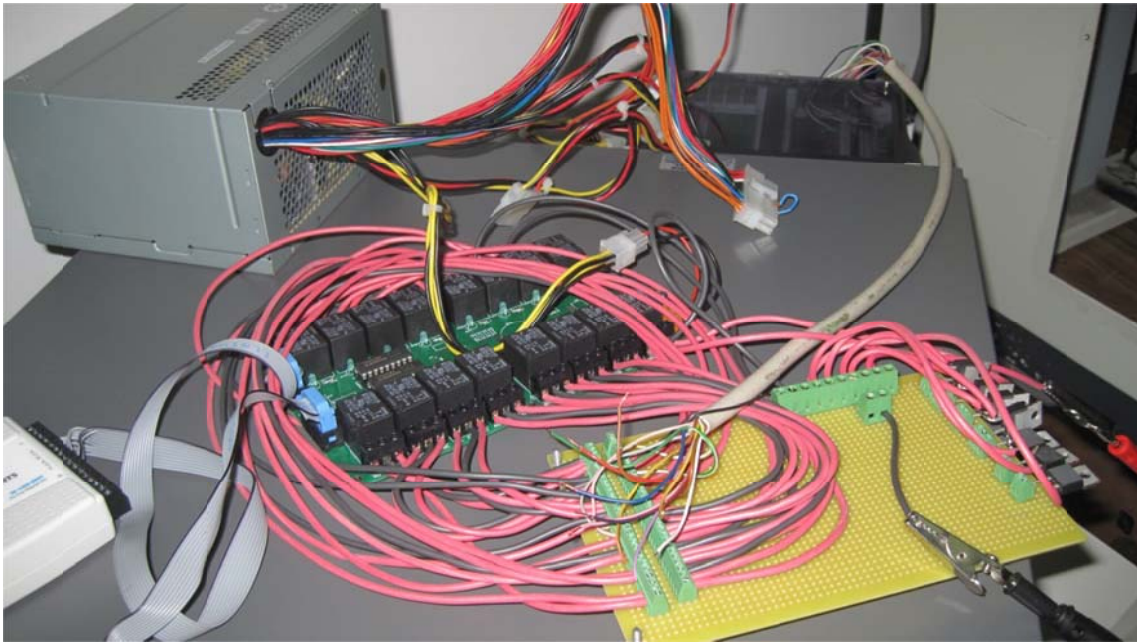


Figura 233: Pruebas

Una vez comprobado el correcto funcionamiento del software y el hardware hasta ese punto, se procede a incluir en el diseño las fuentes de corriente de 1 amperio diseñadas. Y una segunda tarjeta de 16 relés con su DAQ correspondiente, para verificar la correcta comunicación con más de una tarjeta simultáneamente.

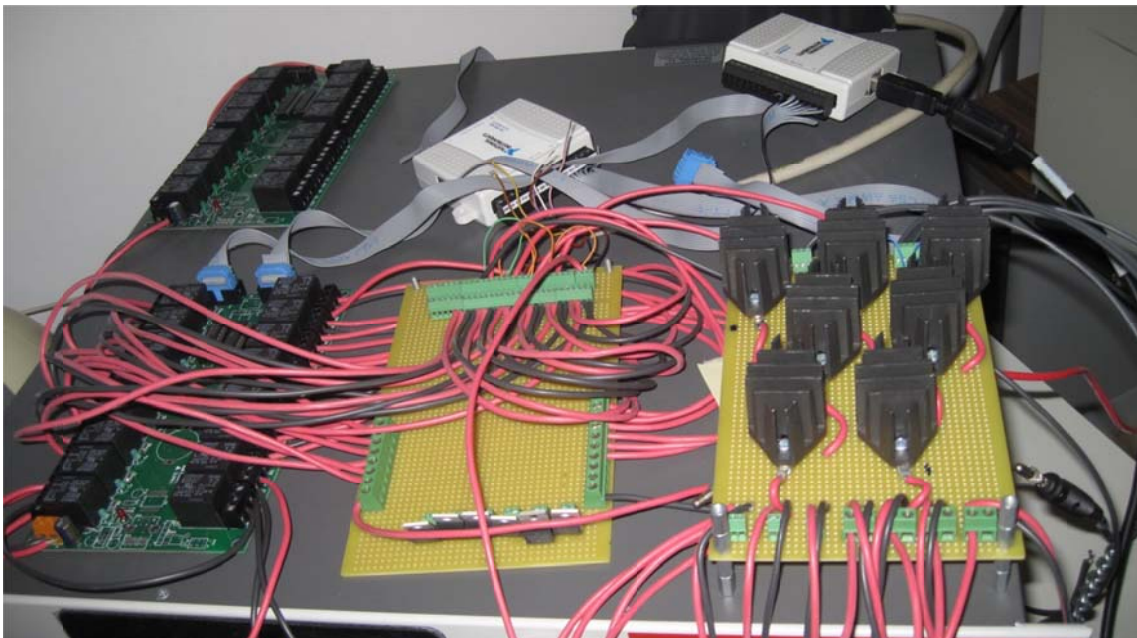


Figura 234: Instalación con dos tarjetas y fuentes de corriente



Figura 235: Detalle potenciómetros fuentes de corriente

10.2. Instalación definitiva

Una vez testeados todos los elementos descritos anteriormente, con el montaje simulado para 1 tarjeta de relés y posteriormente para dos tarjetas de relés con 7 fuentes de corriente de 1 Amperio, así como la aplicación diseñada con labview, se procede a realizar el montaje definitivo de la instalación necesaria para realizar los ensayos acelerados simulando el funcionamiento de la célula solar sometiéndola a un estrés con 3.2 Amperios. Se compraron 25 fuentes de corriente para iniciar los ensayos con 20 células, dado que un futuro se pretende ampliar a 40, e inclusive poner un segundo ensayo en el mismo laboratorio realizando el mismo ensayo modificando el factor de degradación, la temperatura.

Se procede, con la ayuda del personal de IES al montaje e interconexión del circuito, lo primero que se realiza es la distribución del equipamiento para optimizar el espacio disponible, en la imagen se muestra el montaje de la mesa con dos espacios iguales simétricos, el primer montaje se realizará sobre dos bandejas extraíbles, sobre las que irán las tarjetas de relés, los DAQ que las controlan y las fuentes de corriente.



Figura 236: Distribución del espacio

En la primera bandeja extraíble, irán colocadas las 13 primeras fuentes de corriente, controladas con 2 tarjetas de relés, en la primera de ellas irán situadas las 5 primeras fuentes, junto con el control de la alimentación, y en la segunda tarjeta las 8 fuentes de corriente siguientes. Además se aprovecha el espacio para colocar el hub USB.



Figura 237: Bandeja 1

En la segunda bandeja se colocan las 7 fuentes de corriente restantes junto con su tarjeta y DAQ correspondiente, dado que el material recibido eran 25 fuentes de corriente y 4 tarjetas de relés, se opta por dejarlo ya cableado en el sistema, aunque sin ser conectado.



Figura 238: Bandeja 2

En la siguiente figura podemos observar el detalle de las alimentaciones, donde podemos observar la interconexión a un mismo punto de todas las fuentes de corriente.

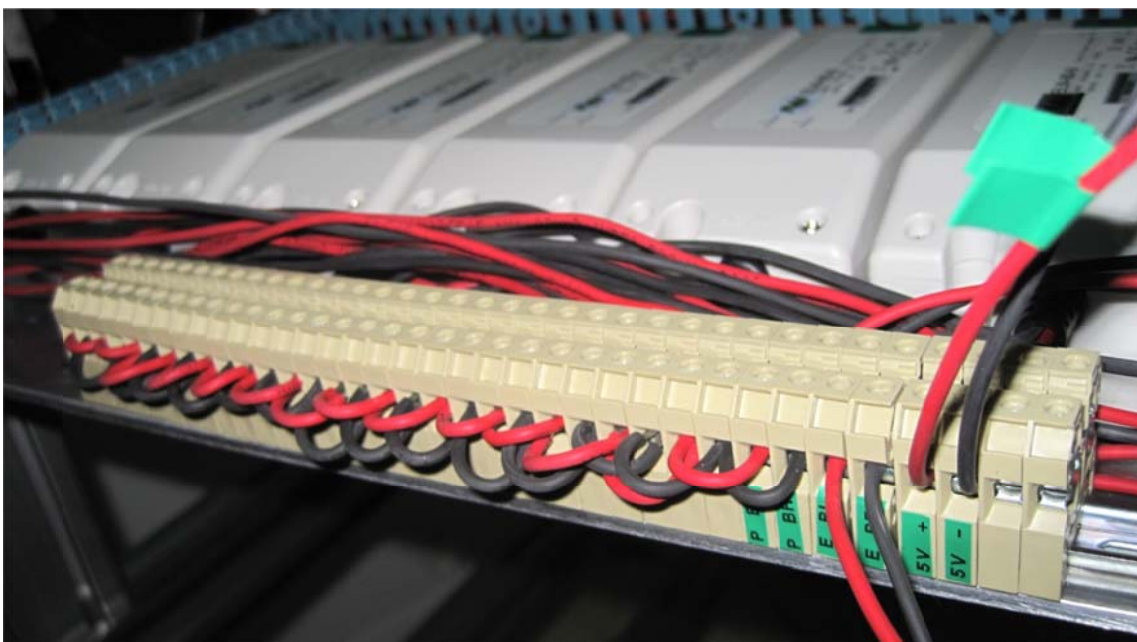


Figura 239: Detalle alimentación



Figura 240: Bandeja 1 y 2

Una vez montadas las bandejas, se procede a cablear las tarjetas de relés con el bornero, el cual corresponderá a 2 hilos de cada célula, que serán los usados para estresarla en el estado de estrés, y posteriormente para inyectarla tensión y medir la corriente con el Agilent. Al bornero se le conectarán además los otros dos hilos de cada célula, éstos irán conectados a la tarjeta del Keithley, y se usarán para la caracterización de la célula mediante su curva IV, midiendo la tensión con el Keithley.

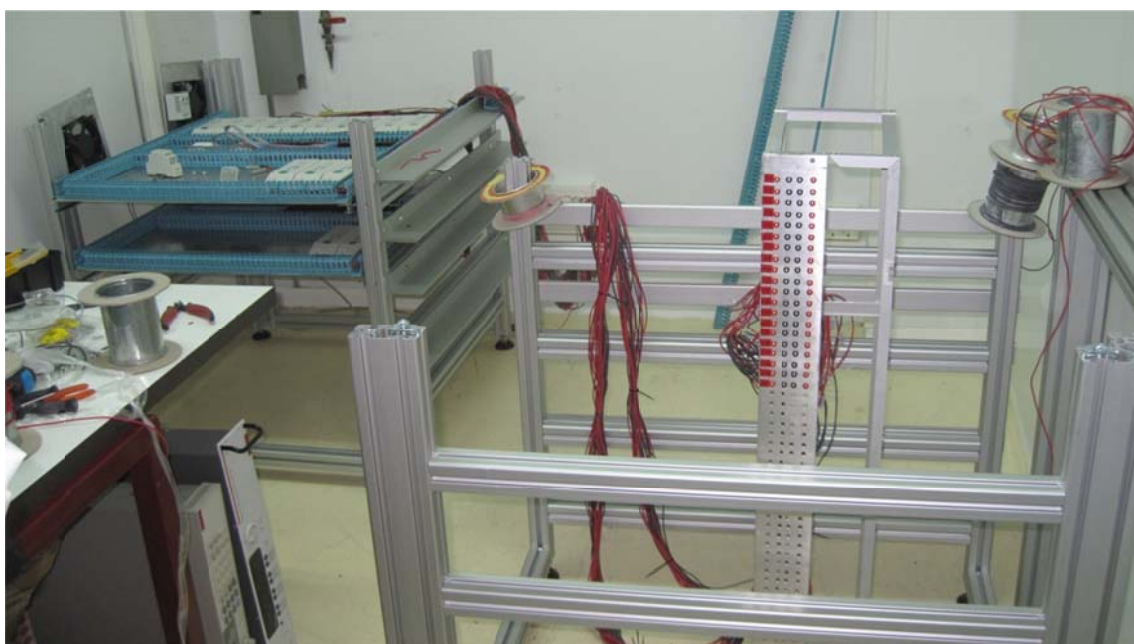


Figura 241: Conexión bandeja relés con bornero

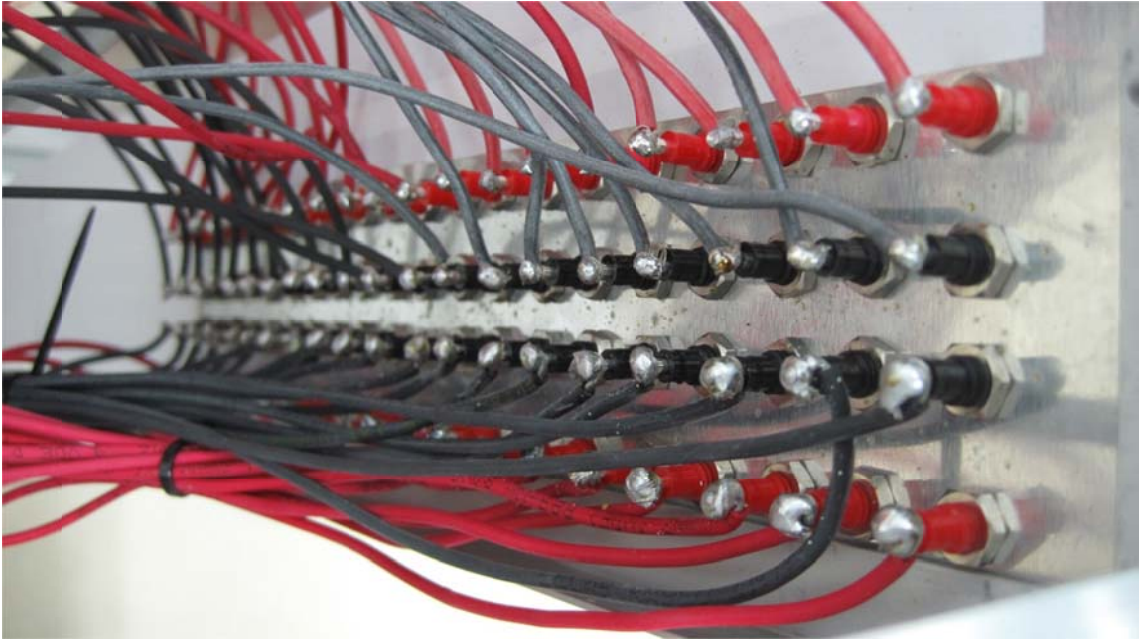


Figura 242: Detalle trasera bornero



Figura 243: Conexión bornero con tarjeta Keithley

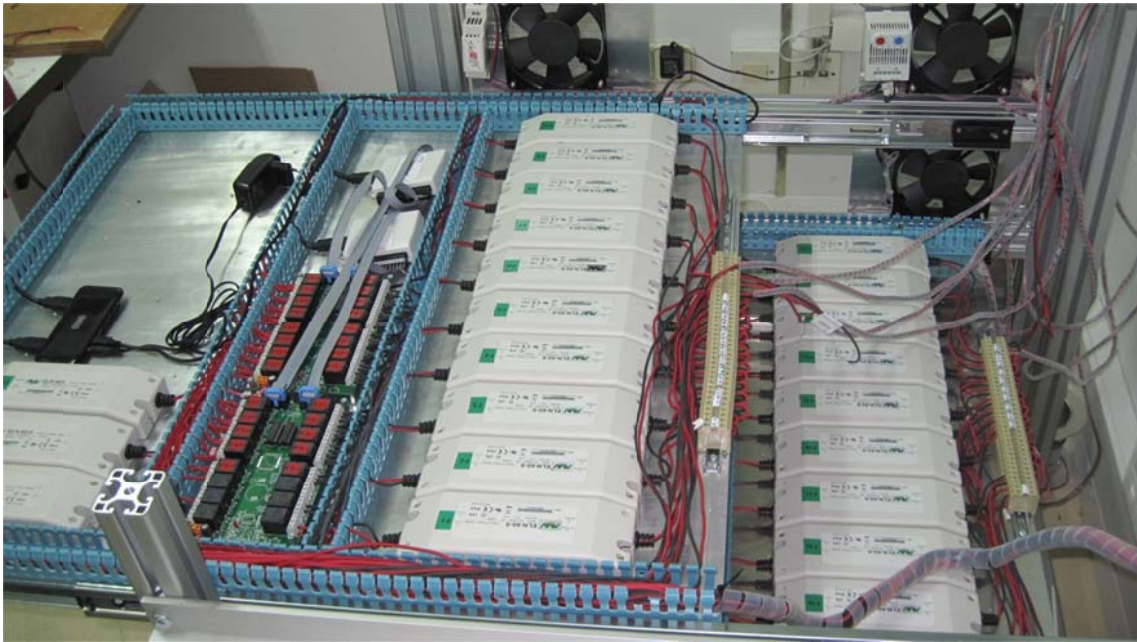


Figura 244: Cableado bandejas



Figura 245: Detalle ventiladores

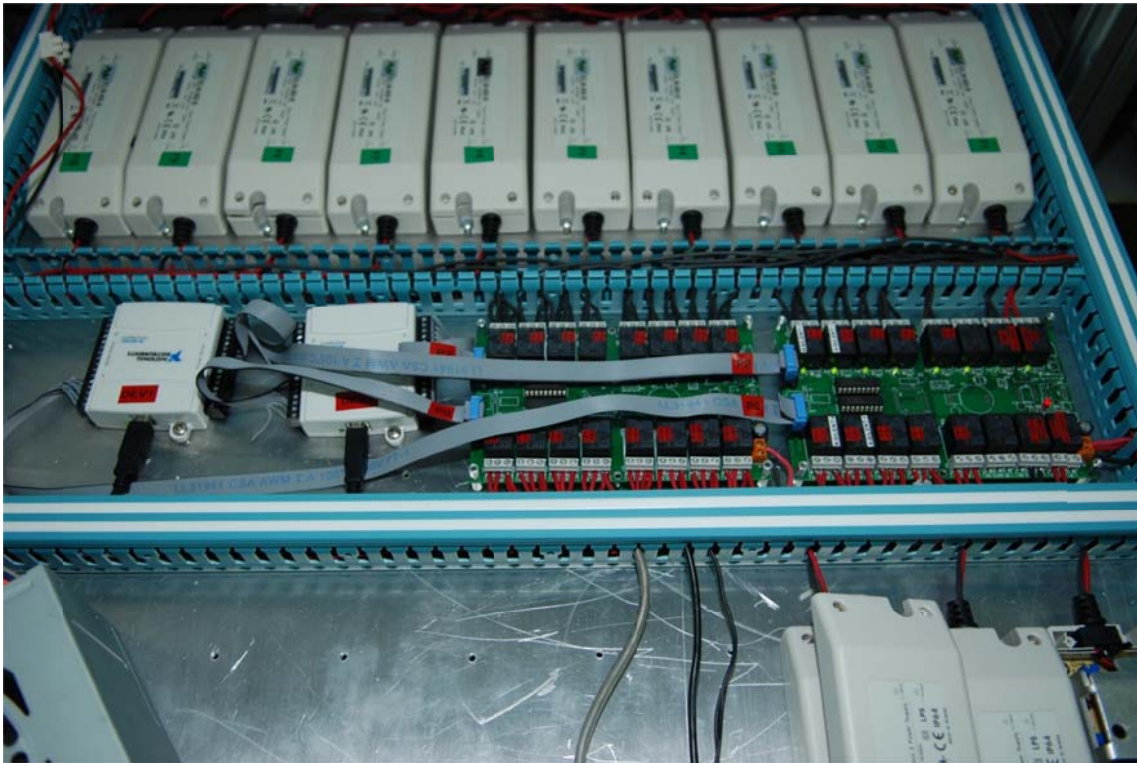


Figura 246: Bandeja 1 funcionando

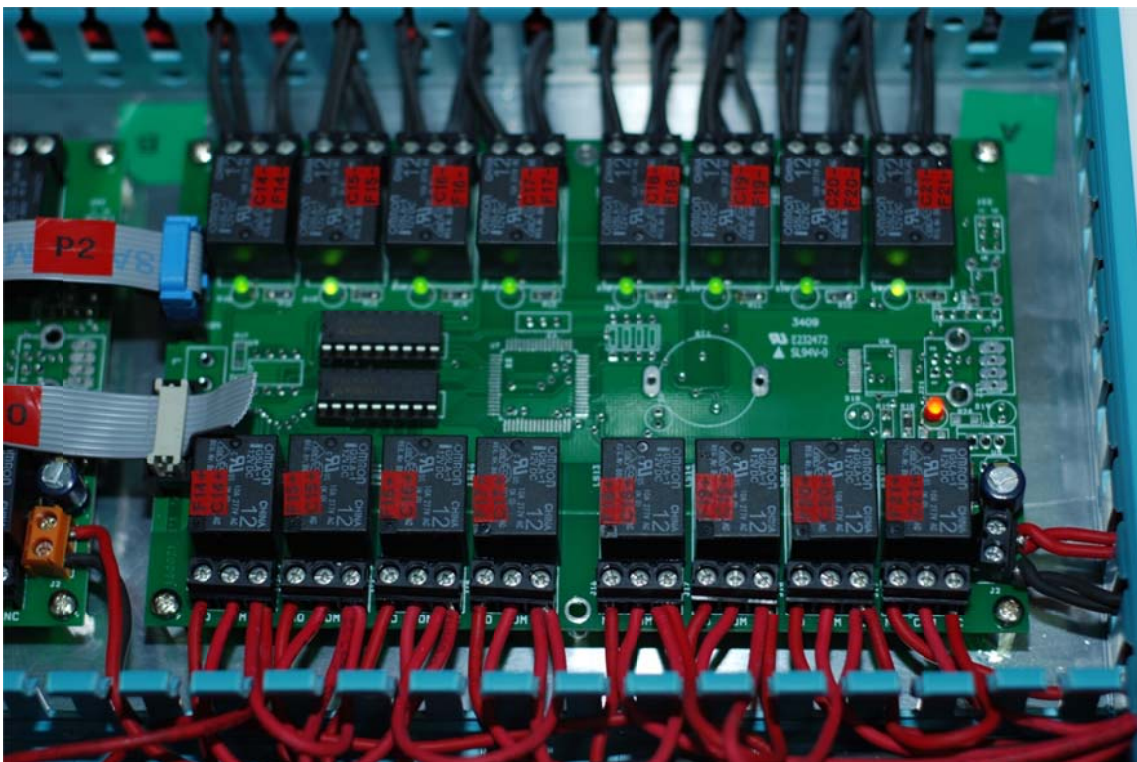
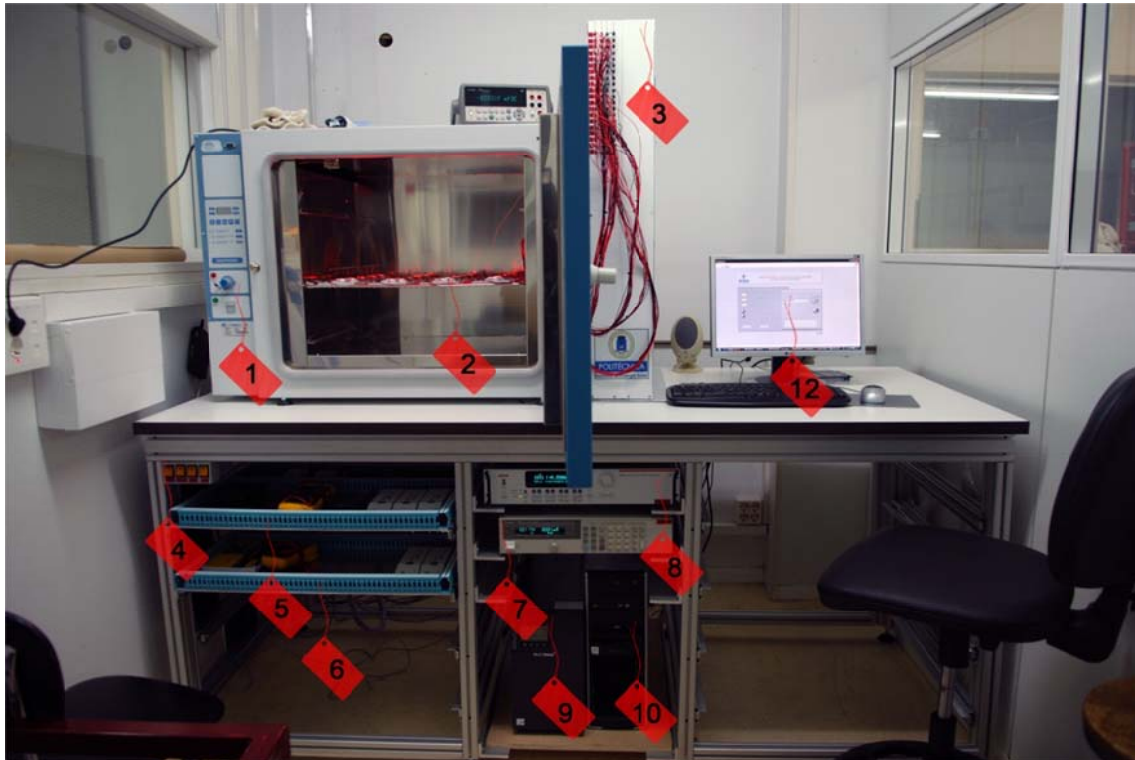
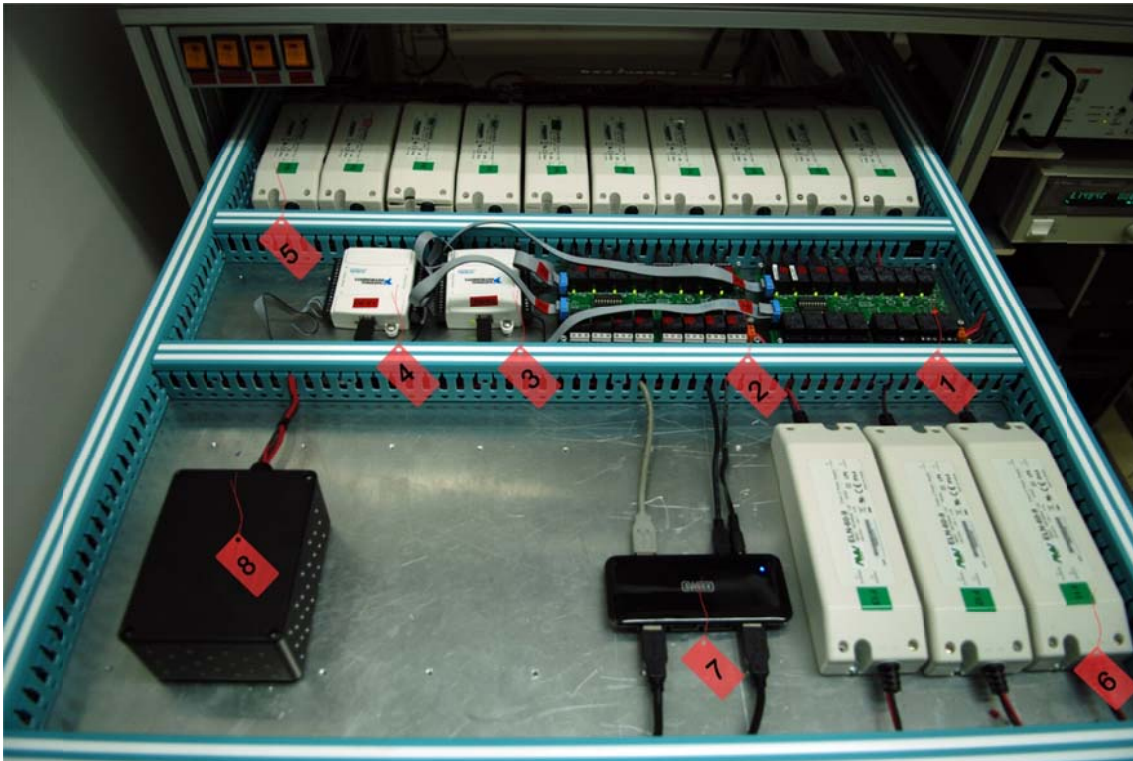


Figura 247: Detalle tarjeta bandeja 2 funcionando

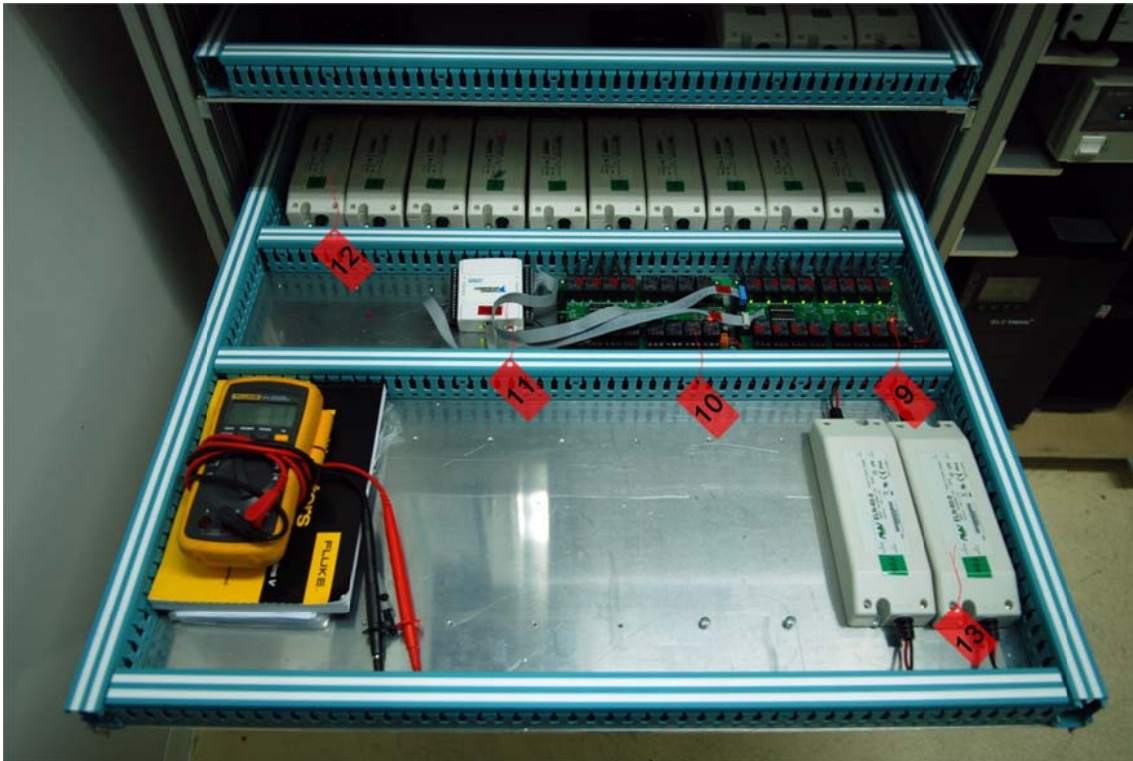
10.3. Esquemas de montaje



1. Estufa universal Digitronic.
2. Células triple unión.
3. Bornero
4. Interruptores alimentación
5. Bandeja 1
6. Bandeja 2
7. Agilent 6631B
8. Keithley 3706
9. SAI
10. Ordenador
11. Software Labview



1. Tarjeta relés 1.
2. Tarjeta relés 2.
3. DAQ dev 1.
4. DAQ dev 2.
5. Fuentes de corriente (1 a 10).
6. Fuentes de corriente (11 a 13).
7. Hub USB.
8. Relés alimentación 220V.



- 9. Tarjeta relés 3
- 10. Tarjeta relés 4
- 11. DAQ dev 3.
- 12. Fuentes de corriente (14-23).
- 13. Fuentes de corriente (24-25).

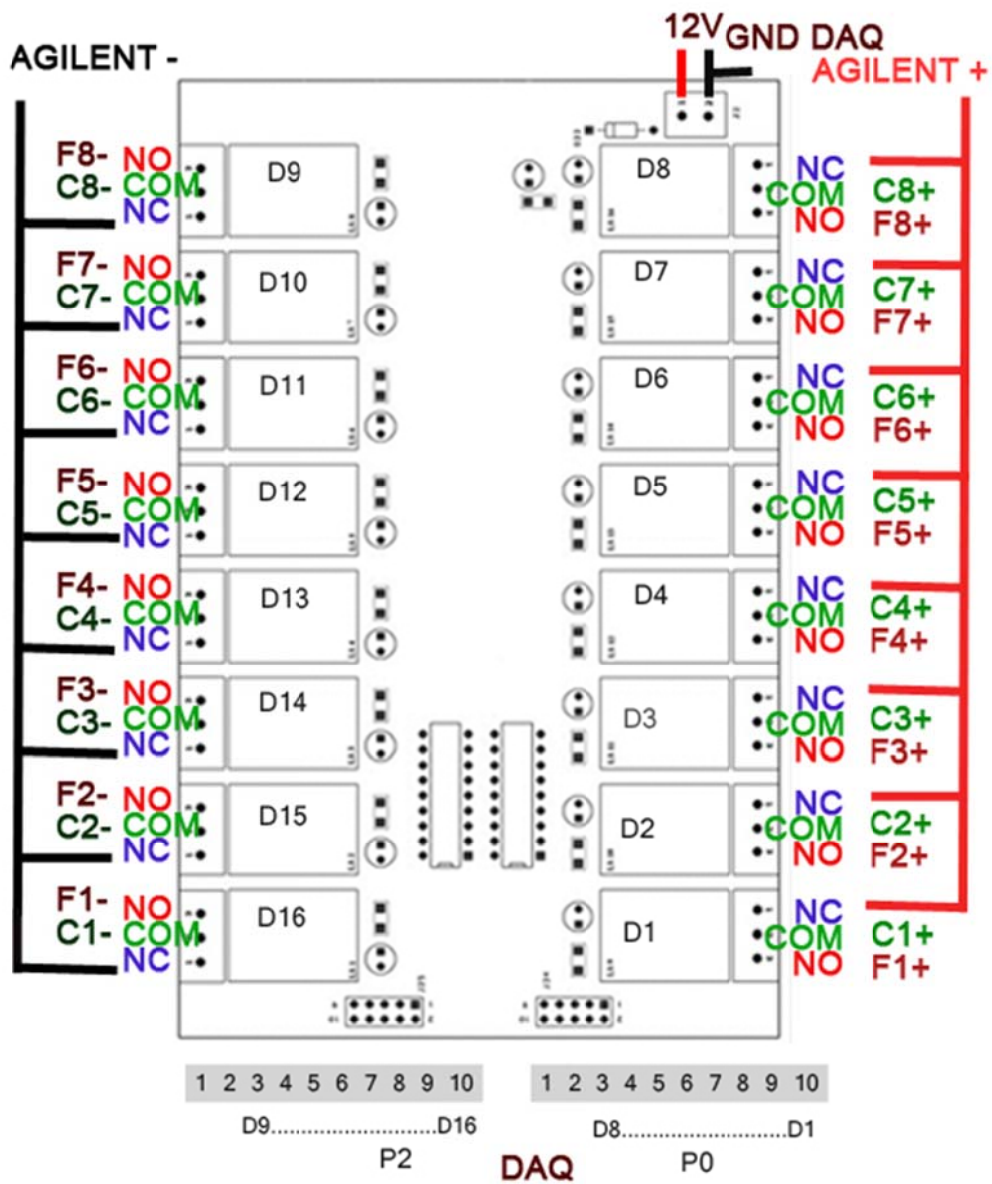


Figura 248: Esquema conexión tarjeta relés (2 a 5)

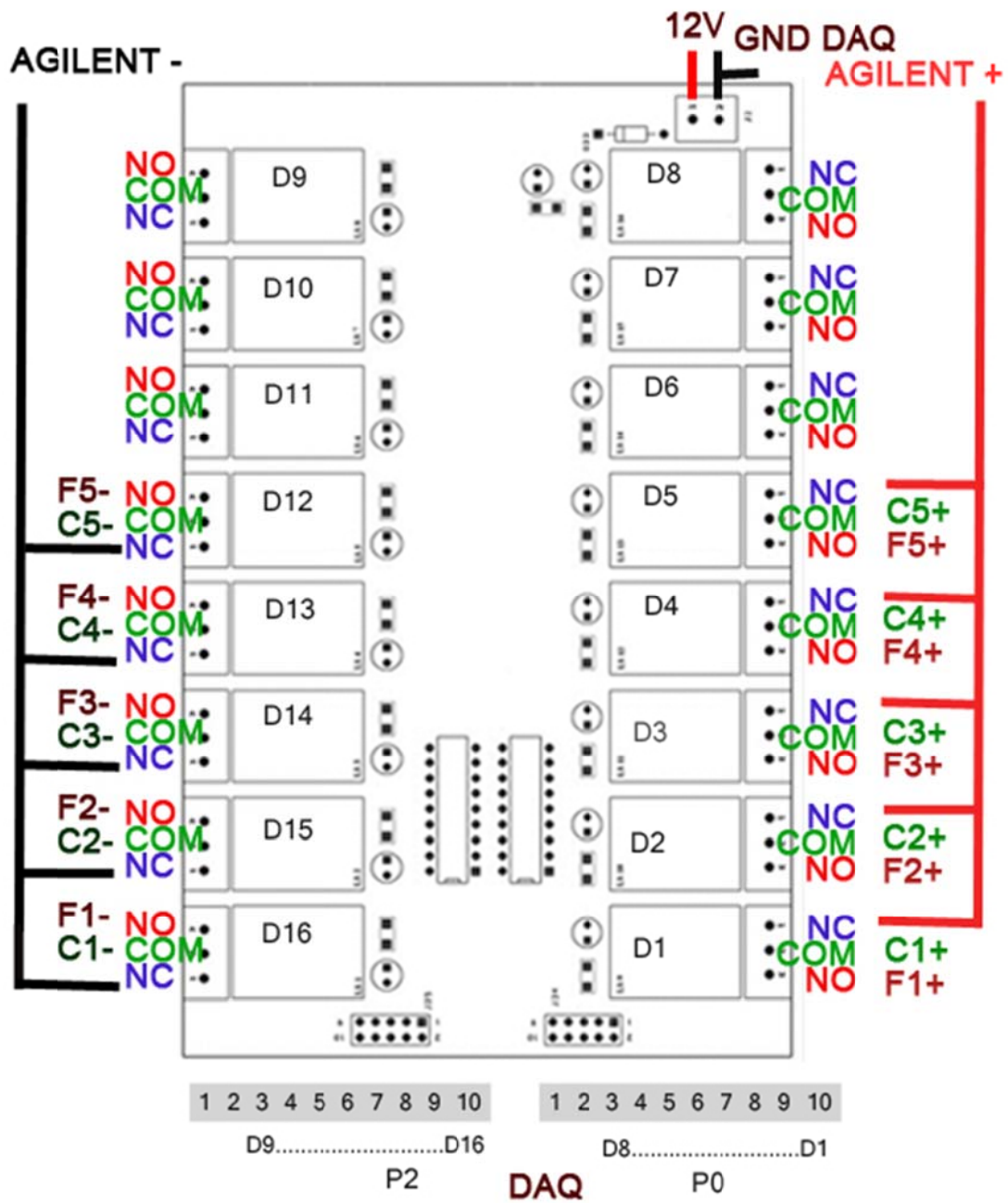


Figura 249: Esquema conexión tarjetas 1 y 6

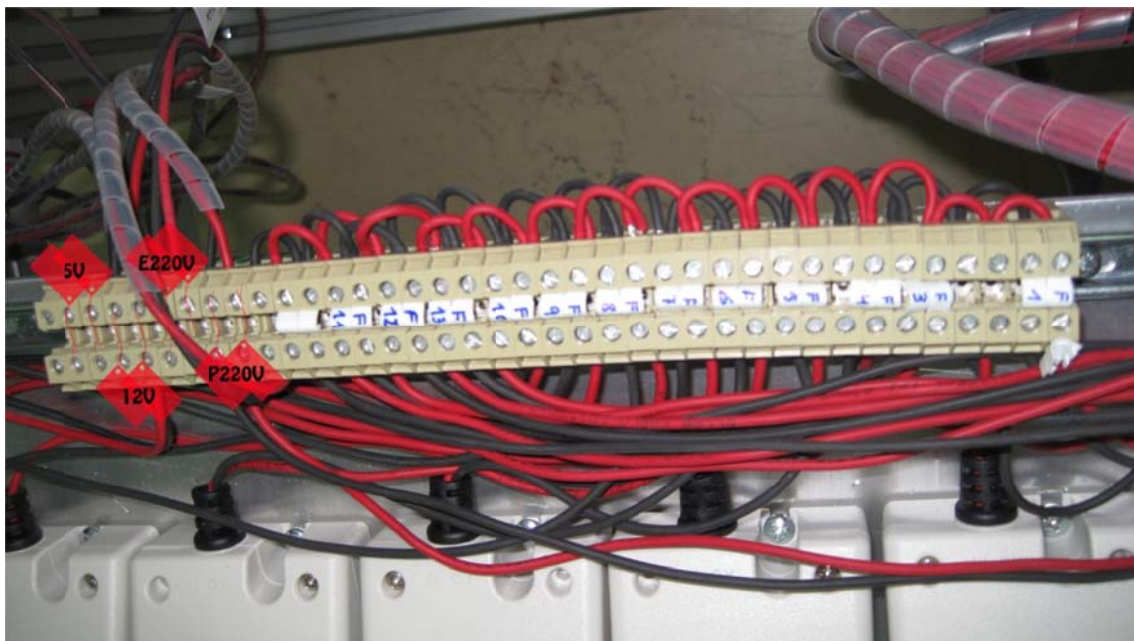


Figura 250 : Alimentación

Conexión de 5V usada para la conmutación de los relés de alimentación, esta conexión pasa por las tarjetas de relés para ser controlada con labview a través del DAQ.

La conexión de 12V usada para la alimentación de las tarjetas de relés.

E220V, indica los cables procedentes del enchufe, el marrón y el azul, esta conexión pasa por los relés de alimentación para seleccionar si se desea que se alimente o no, y vuelve a la conexión P220V, P de placa, de ahí se alimentará a la bandeja número 2 y a todas las fuentes de corriente, las cuales están interconectadas. Primeras medidas con células

Una vez completada toda la instalación y comprobado el cableado, se procedió a verificar la correcta conmutación y el funcionamiento del hardware, con todo el sistema final completo, como ya se ha descrito en los puntos anteriores, usando diodos, se procede a realizar las pruebas con una célula de prueba, estas pruebas consistían en la verificación de la instalación comparando la curva IV obtenida con el sistema descrito con la caracterización de la célula.

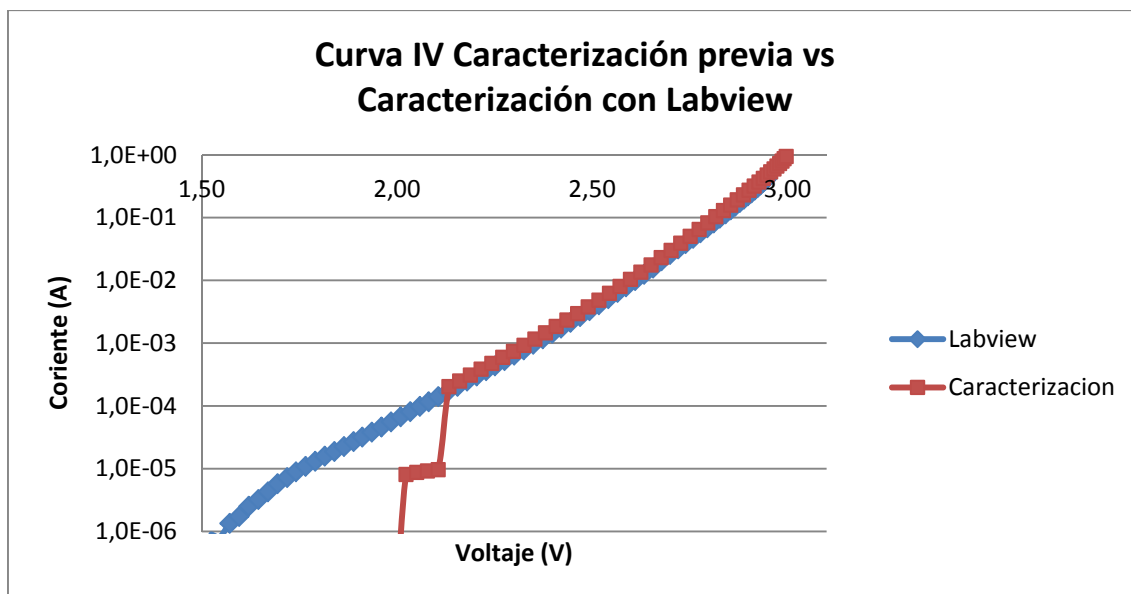


Figura 251 : Curva IV caracterizada vs obtenida con labview

Como se puede observar en la gráfica, no se aprecian pérdidas de corriente por el cableado ni en las conexiones, por lo que el sistema se da por bueno.

A continuación se comprueba la curva con la variación de realizar las medidas inyectando tensión usando el subvi de minMAX o de MAXmin, de tal forma que el cálculo de la tensión a inyectar se calcule desde un valor mínimo indicado a un máximo o viceversa. Dado que se observa que al realizar la medida al inyectar la tensión la célula en frío, la célula se calienta, en la siguiente gráfica se observa la diferencia de realizar una medida u otra, la gráfica de minMAX es fiel a la caracterización, pero llega al límite de la célula con 3,05V, mientras que al comenzar con la tensión máxima, la máxima tensión medida es 3,12V pero se observa el calentamiento provocado desviando ligeramente la curva. Una vez demostrado el comportamiento, se deja a elección del IES el modo en el que deseen realizar los ensayos finales.

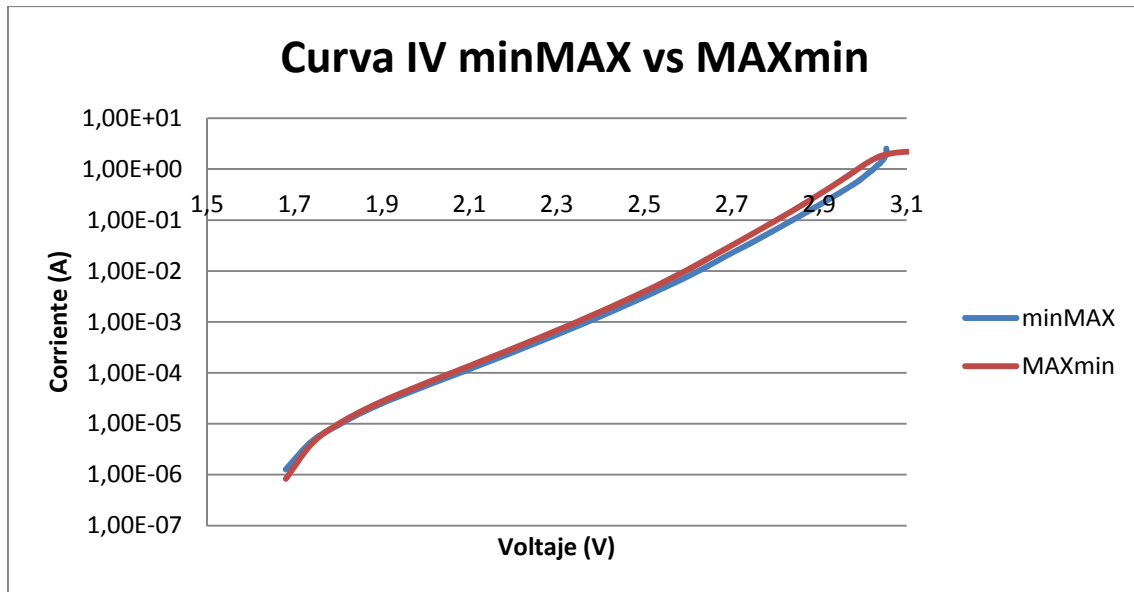


Figura 252: Curva IV minMAX vs MAXmin

La siguiente prueba realizada es el comportamiento de la célula al someterla a un determinado tiempo de estrés, debido al calentamiento de la misma, como podemos observar, es necesario un tiempo de estabilización ya que existen variaciones de la curva I/V debidas a variaciones de la temperatura de la célula.

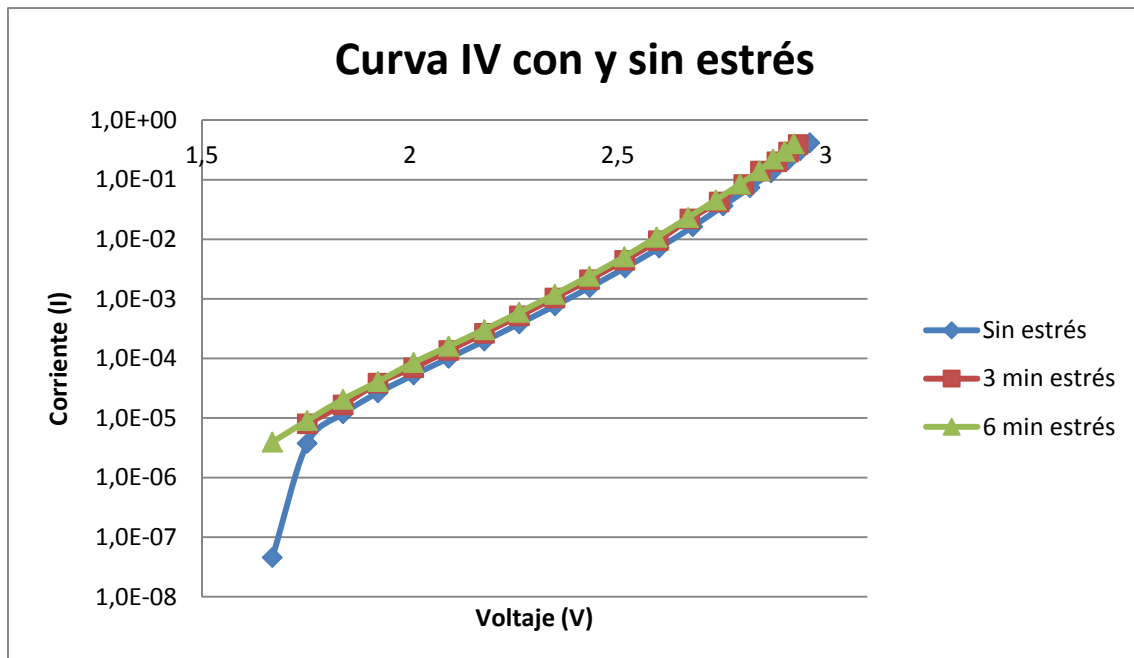


Figura 253: Curva IV con tiempo de estrés

Una vez estimados los valores necesarios para realizar un primer ensayo de verificación de funcionamiento, se estima que los valores de tensión a configurar el Agilent son desde 1.5 a 3.3V, con 2 horas de estrés y 40 minutos de estabilización.

The screenshot shows the 'CONTROL DE TEST' tab of the software. It includes input fields for 'Tensión Mínima' (1.5), 'Tensión Máxima' (3.3), 'Limitación de Corriente' (7), and 'Número de Medidas' (30). There are buttons for 'CARGAR ÚLTIMA CONFIGURACIÓN' and 'GUARDAR CONFIGURACIÓN'. At the bottom, there are fields for 'Tiempo de stress (min)' (0) and 'Numero de Ronda de Medida Actual' (0). On the right, there is a 'Hora actual' field showing '17:10:01', a 'Número de Celula que está midiendo' field showing '0', and a 'Temperatura °C' field. There are also 'INICIO TEST' and 'PARAR TEST' buttons. A 'STOP' button is located at the bottom right.

Figura 254: Configuración primer ensayo

The screenshot shows the 'TEMPORIZACIÓN' tab of the software. It is divided into two main sections: 'TIEMPO DE SIMULACION DE LUZ (STRESS)' and 'TIEMPO DE ESPERA HASTA TOMA DE CURVAS I-V'. The first section has fields for 'Horas (iniciales de simulacion)' (2) and 'Minutos (iniciales de simulacion)' (0). The second section has fields for 'PERIODO Horas (espera hasta medida)' (0) and 'PERIODO Minutos (espera hasta medida)' (40). A 'STOP' button is located at the bottom right.

Figura 255: Tiempo estrés primer ensayo

10.4. Resultados preliminares

Los ensayos acelerados se realizaron en el marco del proyecto de investigación Nueva Generación de Fotovoltaica de Concentración - A new generation of concentrator photovoltaic cells, modules and systems (NGCPV). En este proyecto de investigación Japón-Unión Europea tiene participación de diferentes instituciones universitarias e industriales (UPM, Imperial College, Fraunhofer, ENEA, CEA, Toyota, SHARP,...),

Los primeros ensayos acelerados se han realizado en células solares triple unión de 50 mm² que trabajan entre 400-700 soles. Para poder obtener la ley de vida a condiciones nominales de funcionamiento es necesario realizar los ensayos acelerados de vida a tres temperaturas diferentes, 110°C, 134°C y 154°C. Con tres ensayos acelerados en temperatura es posible evaluar la energía de activación del mecanismo de fallo predominante y a partir de esta la ley, la vida de la célula solar en condiciones nominales de funcionamiento. Para ello se están ensayando tres grupos de 18 células solares en tres cámaras climáticas, cada una de ellas a una de las temperaturas indicadas:

- De las 18 células, 16 de ellas están dentro de la cámara climática con la inyección de corriente que emula las condiciones nominales de trabajo.
- Las otras dos células están dentro de la cámara climática con el objetivo de tener una referencia de la degradación en temperatura sin inyección de corriente. Además una de las dos células se caracteriza mediante las curvas I/V en oscuridad junto a las 16 células con inyección de corriente para poder medir la evolución de la degradación con el tiempo. La otra célula sin inyección de corriente solo se caracteriza mediante las curvas I/V al principio y al final del ensayo con el objetivo de evaluar si la caracterización de las curvas I/V degrada la célula solar.

Los ensayos en las tres cámaras climáticas empezaron en el mes de abril por lo que no están finalizados en el momento de la escritura de la memoria. Sin embargo, del tiempo transcurrido de ensayo sí podemos obtener algunas conclusiones preliminares:

- El sistema de instrumentación, tanto el hardware seleccionado e integrado como el software desarrollado, están funcionando con una

alta fiabilidad. Es de destacar esta alta fiabilidad dada la complejidad del sistema de instrumentación.

- La aceleración de los ensayos aumenta con la temperatura como es de esperar.
 - En el ensayo acelerado a 154°C ya han fallado todas las células por lo que se ha estimado la ley de vida de las células solares a esta temperatura.
 - En el resto de los ensayos todavía está en marcha y en las próximas semanas dispondremos de resultados.
 - A partir de los ensayos acelerados se han observado que las células solares se degradan sin llegar a fallar para al final fallar de forma catastrófica mediante un cortocircuito. Un análisis de fallos del mecanismo de degradación y fallo catastrófico es necesario.

Aunque los ensayos no hayan terminado disponemos de algunos resultados preliminares que los vamos a resumir a continuación.

10.4.1. Mecanismo de degradación y fallo

Para mejorar la fiabilidad de cualquier dispositivo es necesario realizar un análisis del mecanismo de fallo predominante para a partir del análisis estudiar cómo mejorar el dispositivo para que el mecanismo de fallo predominante se mitigue lo más posible. En el estado actual del desarrollo del ensayo solo disponemos de la evolución de las curvas I/V a lo largo del ensayo por lo que no disponemos de todos los resultados necesarios para realizar un análisis de degradación y fallo en profundidad. Sin embargo si podemos realizar un análisis cualitativo de los mecanismos de degradación y fallo de las células solares. En la siguiente figura se observa cómo evoluciona la curva I/V de una célula solar del ensayo a 154°C a medida que pasa el tiempo del ensayo. Se observa que durante las primeras horas se produce una degradación de la curva I/V en la zona de las bajas corrientes, muy lejos de la zona donde se evalúa la potencia de la célula por lo que la degradación de la potencia es mínima. Llegado un momento se produce un cortocircuito en la célula que produce un fallo catastrófico.

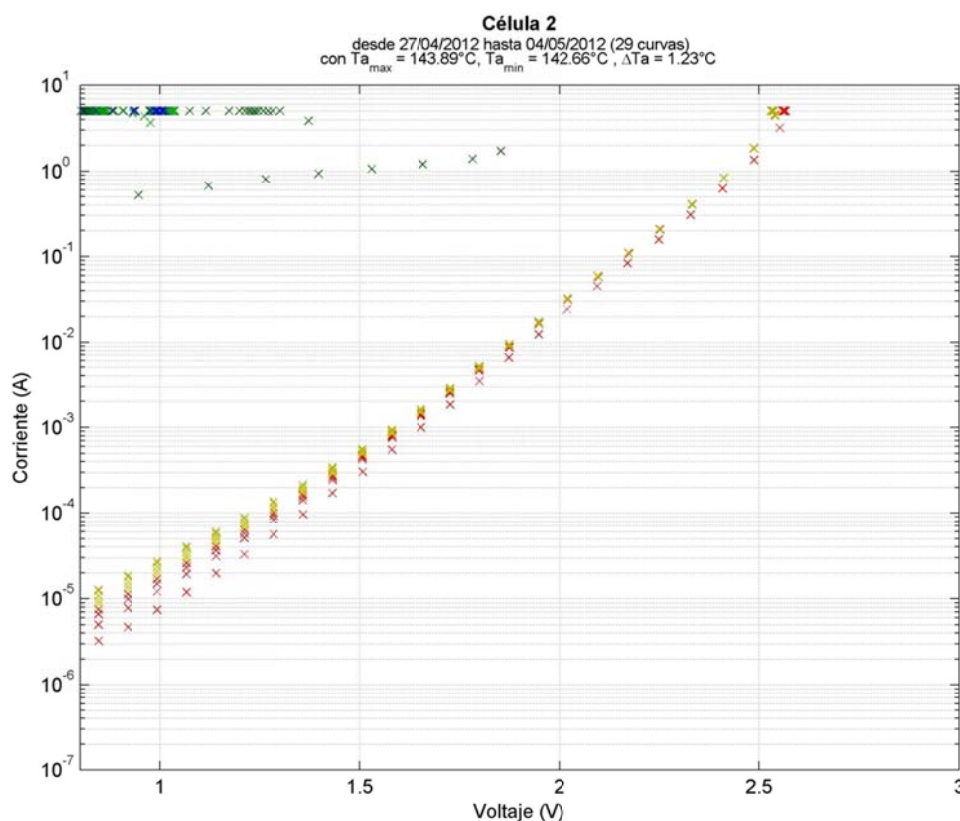


Figura 256: Evolución de la curva I/V de una célula solar durante el ensayo a 154°C .

Para realizar un análisis de mecanismos de fallos pormenorizado es necesario realizar más caracterizaciones de la célula solar antes y después del ensayo.

10.4.2. Análisis preliminar de fiabilidad

Para realizar el análisis de fiabilidad de las células solares es necesario conocer el instante de fallo de los diferentes ensayos, y a partir de estos evaluar la ley de vida del dispositivo a la temperatura del ensayo. Si disponemos de la ley de vida a diferentes temperaturas es posible extrapolarla a la temperatura de funcionamiento mediante modelos de aceleración en temperatura como la ley de Arrhenius.

En la actualidad sólo disponemos de los tiempos de fallo del ensayo en temperatura más acelerado, 154°C , por lo que podemos evaluar la ley de vida a esta temperatura. Cuando finalicen los otros dos ensayos podremos evaluar mediante la ley de vida a temperatura nominal de funcionamiento mediante la ley de Arrhenius.

Los tiempos de fallo del ensayo a 154°C son los siguientes:

Célula	t fallo	fecha fallo	
1	14	28-abr	16:13
2	18	28-abr	22:43
3	32	30-abr	10:27
4	86	07-may	15:55
5	40	01-may	19:53
6	28	30-abr	4:07
7	28	30-abr	4:08
8	BUENA (S/I)		
9	32	30-abr	10:32
10	28	30-abr	4:10
11	38	01-may	16:49
12	36	30-abr	17:03
13	22	29-abr	18:45
14	BUENA (S/I)		
15	14	28-abr	16:22
16	20	29-abr	15:38
17	16	28-abr	19:38
18	16	28-abr	19:39

Tabla VII: Tiempo de fallo de las diferentes células del ensayo a 154°C (las células 8 y 14 son las células sin inyección de corriente).

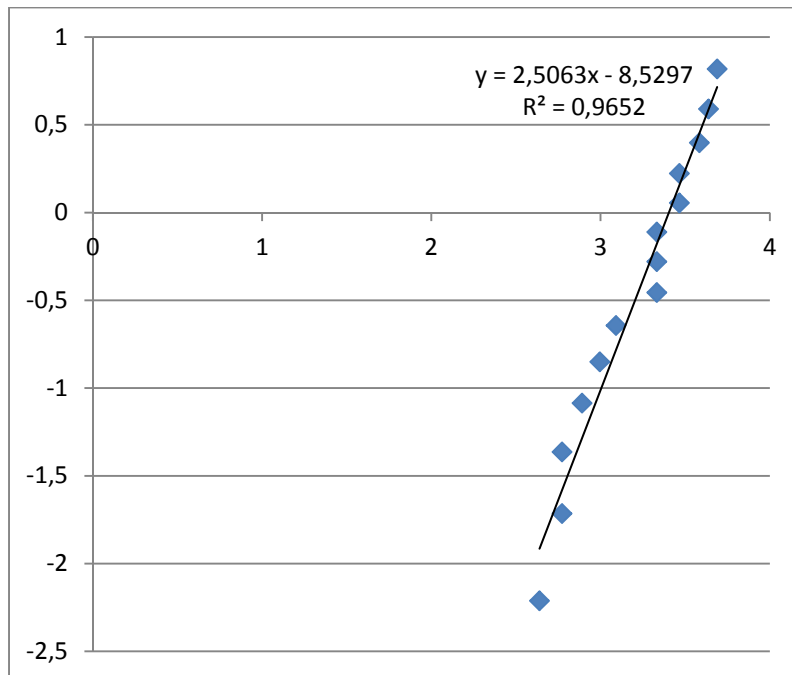


Figura 257: Representación de Weibull de los tiempo de fallos

- A partir de los datos de tiempos de fallo es posible evaluar la ley de vida de las células solares a 154°C mediante la ley de Weibull. Para ello linealizaremos la ley de Weibull y representaremos los datos en representación de Weibull ($\ln\left(\ln\left(\frac{1}{1-F}\right)\right)$ vs $\ln t$). Los valores de $F(i,n)$ se calculan mediante rangos medianos $F(i,n) = \frac{i-0,3}{n+0,4}$ los parámetros de Weibull ($\beta=2,5063$ y $\eta=30,06$ horas). De esta forma la fiabilidad a 154°C sigue la siguiente función, $R(t) = e^{-\left(\frac{t}{30,06}\right)^{2,5063}}$.
- Hasta que no finalicen los otros dos ensayos no es posible obtener la ley de vida a condiciones nominales de funcionamiento.

Sin embargo, lo más importante a destacar es que el sistema de instrumentación cumple plenamente las especificaciones planteadas con un alto nivel de fiabilidad.

11

Conclusiones

Dada la escasez de recursos energéticos y el precio de los mismos, cada día se hace más necesario explotar la única fuente de energía inacabable que disponemos, el sol. Para ello es necesaria la instalación de paneles solares, sistemas fotovoltaicos que conviertan la energía del sol en energía eléctrica.

Cuanto más eficaces sean los sistemas fotovoltaicos, mayor será la energía obtenida, pero ahí encontramos varios problemas, construir un sistema fotovoltaico es caro. Inicialmente se comenzaron construyendo células de silicio con una única unión, aparentemente “baratas” de fabricar, pero con una eficiencia en módulo del orden del 16% y una durabilidad que ronda los 25 años. Estas prestaciones se pueden mejorar, y en ello están los distintos centros de investigación de todo el mundo, entre ellos el Instituto de Energía Solar de la Universidad Politécnica de Madrid. Una de las áreas de investigación más activas es la del desarrollo de células solares con nuevos materiales como las células solares multiunión con semiconductores III-V. Uno de los principales problemas de estas células es que son mucho más caras de fabricar que las células de silicio, de ahí que se hayan implementado en 1mm^2 , usando un concentrador solar, de tal forma que el sol recibido por la célula de 1mm^2 es equivalente a 1000 soles. Células de este tamaño y nivel de concentración han alcanzado eficiencias del 43%., Aunque la alta fiabilidad de este tipo de células se ha demostrado en aplicaciones espaciales sin concentración, es necesario validarlas para aplicaciones terrestres con alto nivel de concentración.

Por todo esto, es por lo que surge la necesidad de evaluar la fiabilidad de las células solares multiunión III-V, y para ello en este proyecto se ha desarrollado el software necesario para llevar a cabo dicho estudio. La metodología para realizar ensayos acelerados en células solares, cuenta con una peculiaridad, éstas deben simular las condiciones de funcionamiento dentro de una cámara climática, para ello, había varias opciones, en este proyecto se ha optado por emular las condiciones de radiación someténdolas a una corriente equivalente al estrés generado cuando la célula está en funcionamiento, usando como factor de aceleración la temperatura de la cámara climática. Las células estarán sometidas a un estrés durante un tiempo determinado, que se interrumpirá de forma periódica para caracterizar la célula con su curva IV y se volverá a someter a estrés.

Con este proyecto se pretende poder estudiar la fiabilidad de las células solares, no sólo su rendimiento, sino la durabilidad y su robustez y los motivos de fallo si los hubiera.

A la finalización de este proyecto, en el mes de abril, se ha dejado instalado y funcionando el ensayo con 20 células,



Figura 258: 20 células funcionando.

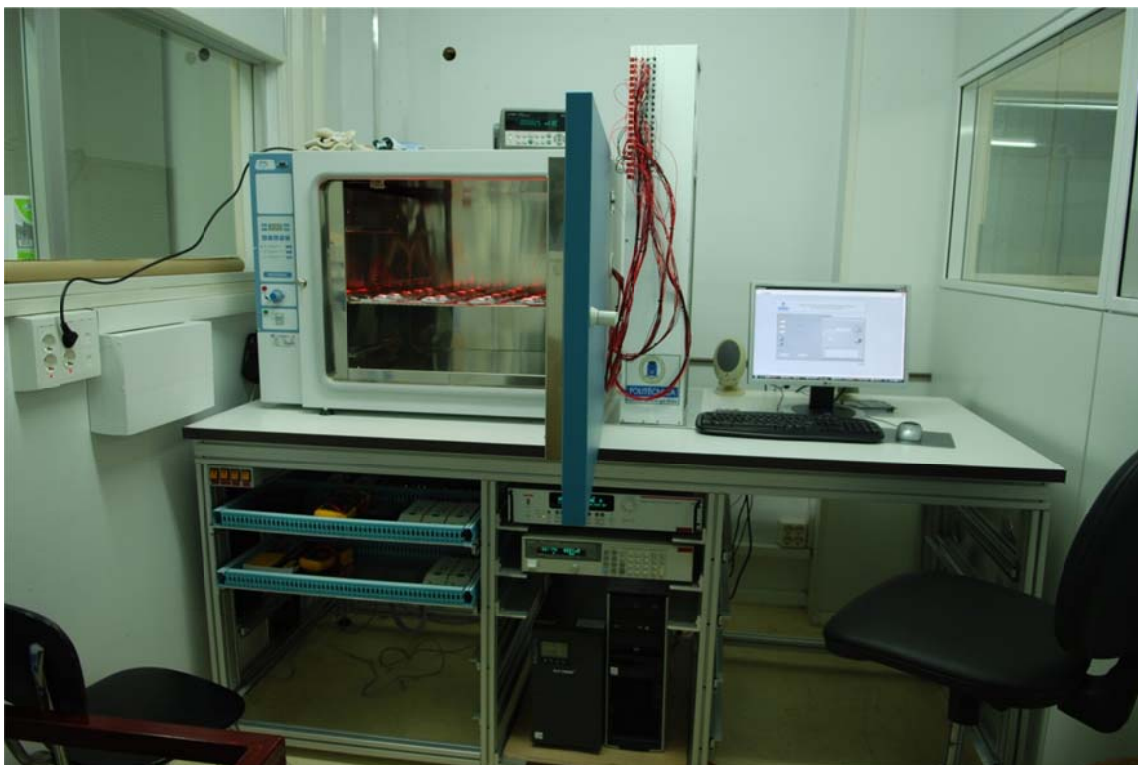


Figura 259: Primer ensayo 20 células

A día de hoy, en mayo, hay tres ensayos acelerados a tres temperaturas diferentes, 110°C, 134°C y 154°C, ya están funcionando y los primeros resultados han sido obtenidos:

- El ensayo a mayor temperatura, 154°C, es el de mayor aceleración y el único que ha finalizado hasta el momento. A partir de los datos de tiempos de fallo de este ensayo se ha evaluado la ley de vida de la célula solar a esta temperatura. Del análisis de Weibull de los tiempos de fallo de este ensayo se concluye que los parámetros de Weibull son $\beta=2,5063$ y $\eta=30,06$ horas. De esta forma la fiabilidad a 154°C sigue la siguiente función, $R(t) = e^{-\left(\frac{t}{30,06}\right)^{2,5063}}$.
- Para evaluar la fiabilidad del dispositivo en condiciones nominales de funcionamiento es necesario concluir los otros tres ensayos en marcha, 100°C y 134°C.
- Dado que los ensayos no han sido finalizados no se puede realizar un análisis exhaustivo del mecanismo de fallos de las células pero sí que se puede adelantar que los fallos catastróficos se producen por cortocircuito y previamente al fallo catastrófico se produce una degradación en la zona de bajas corrientes de la curva I_V de las células solares.

Estos resultados preliminares de los ensayos muestran que el sistema de instrumentación desarrollado en este Trabajo Fin de Master cumple las especificaciones planteadas en un principio con un alto nivel de fiabilidad.

12 Bibliografía

- Algora, C., Ortiz, E., Rey-Stolle, I., Díaz, V., Peña, R., Andreev, V. M., y otros. (2001). A GaAs Solar Cell with an Efficiency of 26.2% at 1000 Suns and 25.0% at 2000 Suns. *IEEE TRANSACTIONS ON ELECTRON DEVICES*, VOL. 48, NO. 5, MAY 2001.
- Algora, C., Rey-Stolle, I., García, I., Galiana, B., Baudrit, M., Espinet, P., y otros. (2009). III-V Multijunction solar cells for ultra-high concentration photovoltaics. *IEEE*, 1571-1575.
- Algora, C., Rey-Stolle, I., García, I., Galiana, B., Espinet, P., M., B., y otros. (Nov. 2008.). A dual junction solar cell with an efficiency of 32.6% at 1000 suns and 31.0% at 3000 suns. *5th Intern. Conf. for the Generation of Electricity*. Palm Desert.
- Antón, I. (2004). *Métodos y equipos para la Caracterización de Sistemas Fotovoltaicos*. Tesis Doctoral UPM.
- Antón, I., Sala, G., Heasman, K., Kern, R., & Bruton, T. (30 JAN 2003). Performance prediction of concentrator solar cells and modules from dark I-V characteristics. *Prog. Photovolt: Res. Appl.*, 11: 165–178. doi: 10.1002/pip.477.
- Barrera, M. (2009). Simulación y caracterización de celdas solares multijuntura y de silicio cristalino para aplicaciones espaciales. Tesis Universidad Nacional de General San Martín Comisión Nacional de Energía Atómica.
- Belmonte, E. (s.f.). Instrumentación Ensayos acelerados.
- E. Barrigon, I. R.-S. (2009). GaInP/GaInAs/Ge triple junction solar cells for ultra high concentration. *IEEE*, 383-386.
- E. García-Tabarés, I. G.-S. (2011). Integration of III-V Materials on Silicon Substrates for Multi-junction Solar Cell Applications. *IEEE*, Proceedings of the 8th Spanish Conference on Electron Devices, CDE'2011.
- Enrique Barrigón, E. G.-T.-S. (2011). Triple-Junction Solar Cells for Ultra-High Concentrator Applications. *Proceedings of the 8th Spanish Conference on Electron Devices, CDE'2011*.

- Enrique Barrigón, E. G.-T.-S. (2011). Triple-Junction Solar Cells for Ultra-High Concentrator Applications. *IEEE, Proceedings of the 8th Spanish Conference on Electron Devices, CDE'2011*.
- Galiana, B., Garcia, I., Gonzalez, J., Baudrit, M., Rey-Stolle, I., & Algora, C. (2007). III-V multijunction solar cells for concentrations around 1000X: the IES-UPM strategy. *IEEE*, 242-245.
- Ignacio Rey-Stolle, C. A. (2009). Simulating III-V concentrator solar cells: A Comparison of advantages and limitations of lumped analytical models; Distributed analytical models and numerical simulation. *IEEE*, 1622-1627.
- Instrument, N. (s.f.). *www.ni.com*.
- J. R. González, C. A.-S. (2006). Strategy for Certified Reliability Analysis of III-V High Concentration Solar cells. *IEEE*, 702-705.
- Linio, S. (2008). *Ensayos acelerados de componentes pasivos*. Proyecto Fin de Carrera EUITT UPM.
- N. Nunez, M. V. (2010). Novel accelerated testing method for III-V concentrator solar cells. *ESREF*.
- Núñez, N. (febrero de 2012). Ensayos acelerados de fiabilidad de células solares de concentración. Tesis doctoral UPM.
- Vázquez, M., & Jiménez, F. J. (2009). Confiabilidad de Sistemas. *Asignatura Máster Ingeniería de Sistemas y Servicios Accesibles para la Sociedad de la Información*. EUITT UPM.
- Vázquez, M., Narvarte, L., & Antón, I. (2010). Ingeniería fotovoltaica. *Asignatura Máster en Ingeniería de Sistemas y Servicios Accesibles para la Sociedad de la Información*.
- Wells, L. K., & Travis, J. (September 26, 1996). *LabVIEW for Everyone: Graphical Programming Made Even Easier*. National Instrument.
- www.weibull.com*. (s.f.).

